Métodos de Desenvolvimento de Software (MDS) 2014/2015

Vasco Amaral, Miguel Goulão
vma@fct.unl.pt, mgoul@fct.unl.pt
http://moodle.fct.unl.pt/MDS201415

Labs 1, 2

Software Project Planning

2

Project planning

Software project planning

- Identify activities
- Schedule activities
- Assign resources

Activity identification

Work Breakdown Structure

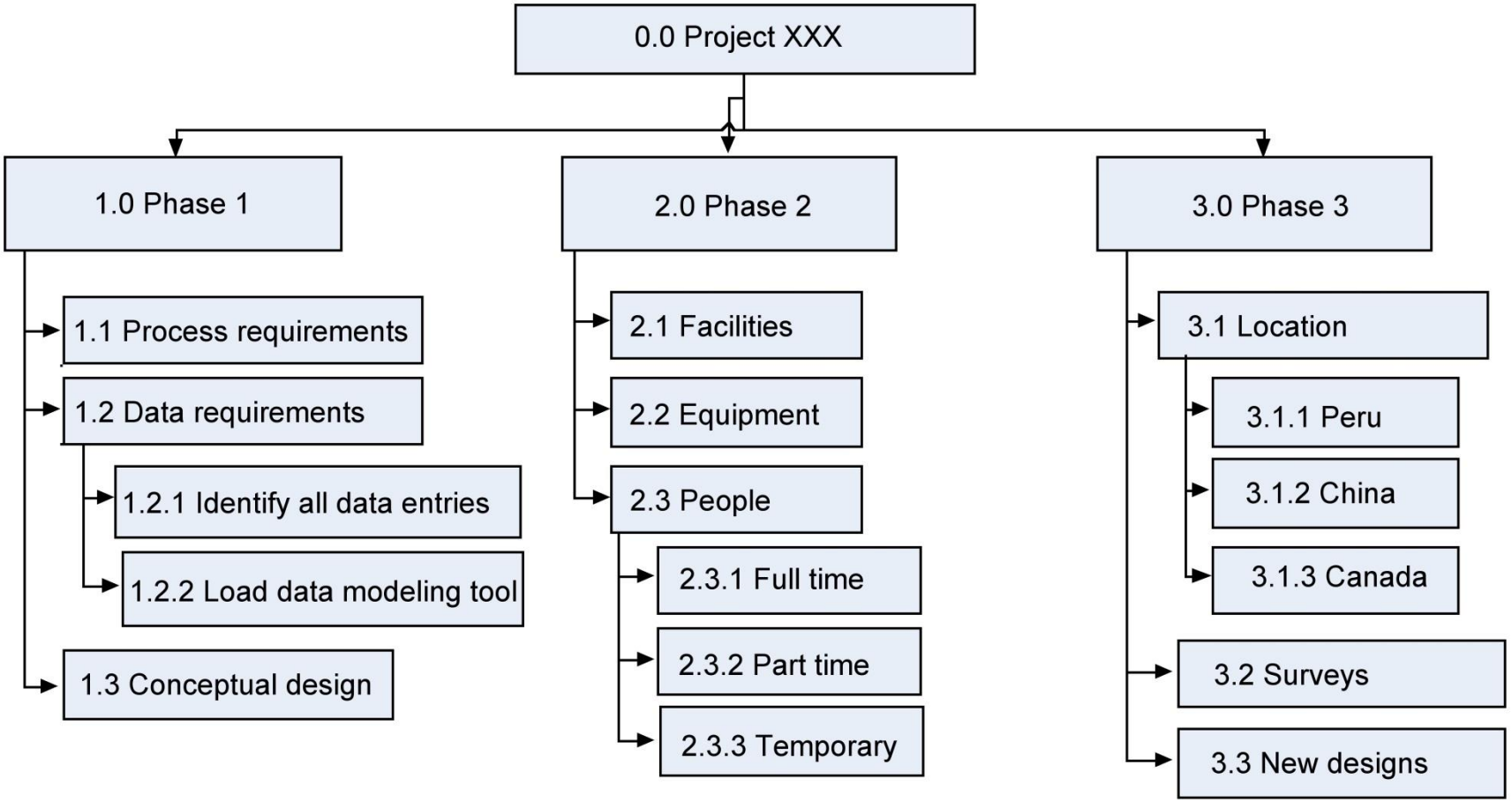
Work Breakdown Structure (WBS)

- Defines the scope of the project (“to do” list)
- Breaks work down to components
  - Subdivides complex tasks into simpler ones
- Hints on building a WBS
  - Include 100% of the work / deliverables
  - Avoid overlapping elements
  - Plan first for work outcomes, rather than actions
  - Try to get the “sweet spot” of detail
    - Too little makes planning harder
    - Too much hinders the communication role of the WBS

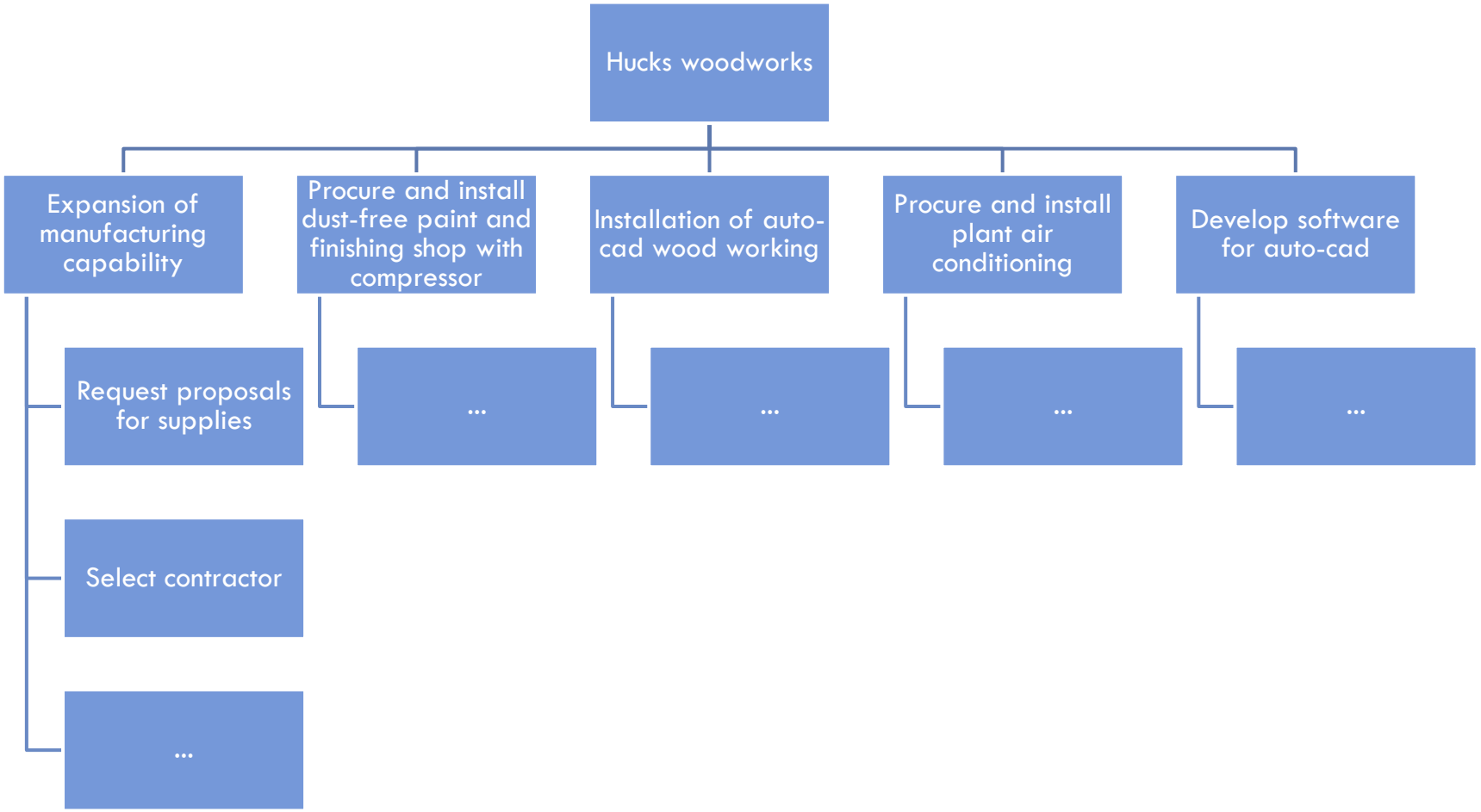
Creating WBSs

- Several possible decomposing strategies
  - Product oriented
  - Process oriented
  - ...
- The WBS may have a different number of levels
  - Enough to facilitate estimates on costs, resources, ...
  - Not too many, to facilitate communication
- The lower level components are **work packages**
  - Must be assigned to individuals, or teams, responsible for delivering them
  - Estimates of time, costs and resources are done at the work package level of granularity

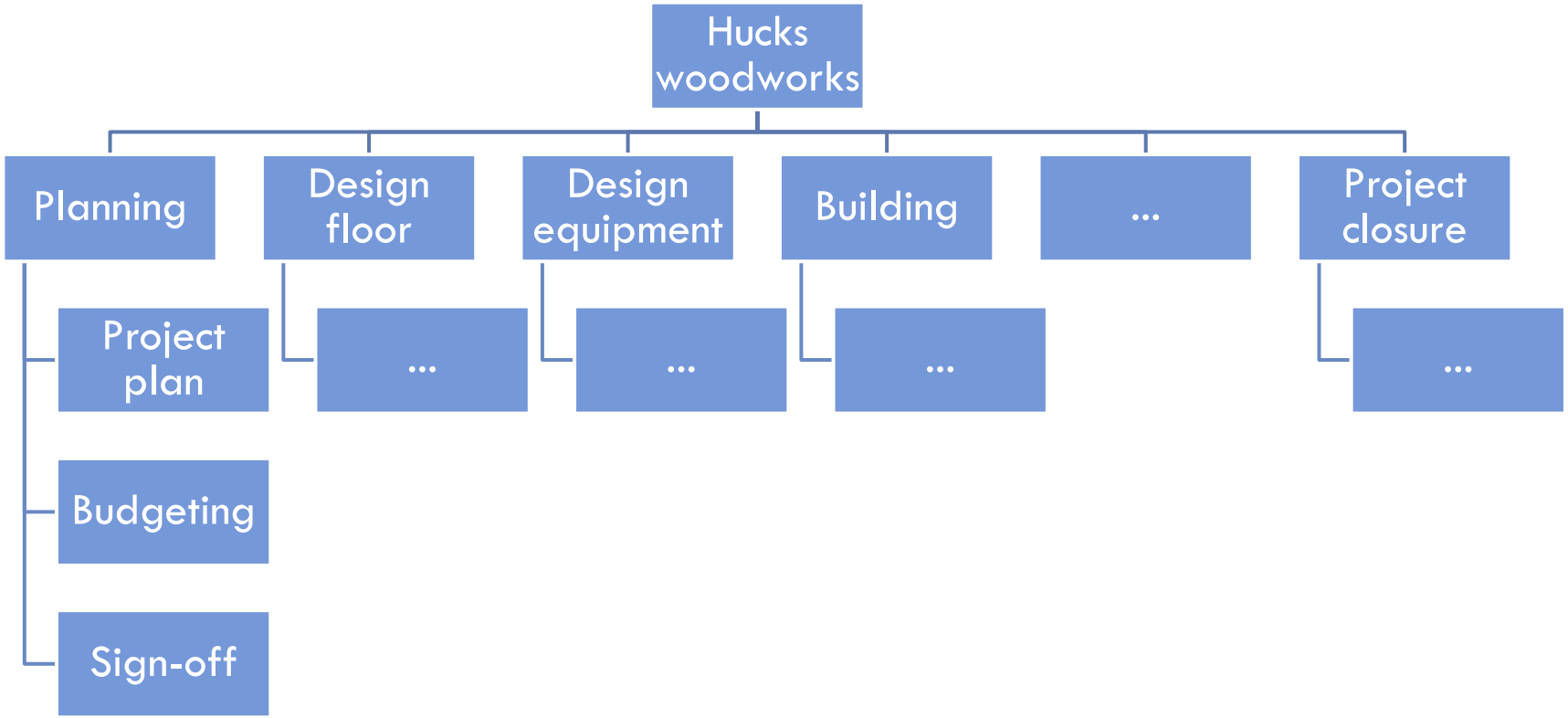
Example: Project XXX WBS



Hucks woodworks (product oriented)

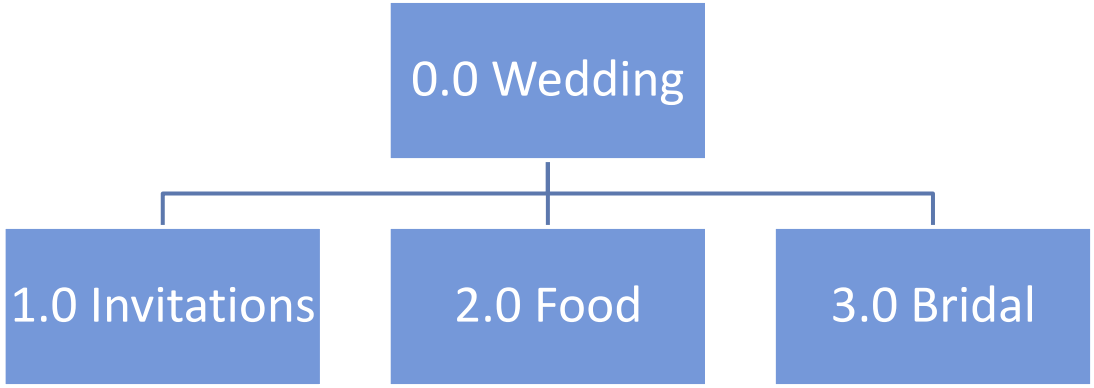


Hucks woodworks (process oriented)



Peggy Sue is getting married! Uhuuuu!

- But of course, she is in panic. Help her planning her **wedding**.
- Sort these activities out under **Invitations**, **Food**, or **Bridal**:
  - Shop for shoes
  - Create guest list
  - Tailoring and fitting
  - Shop for dress
  - Find caterer
  - Cater the wedding
  - Wait for RSVPs
  - Mail the invitations
  - Finalize the menu
  - Print the invitations
  - Choose the bouquet



Exercise: Multimedia edition project

- Your music band is launching a multimedia project that will include a book, a cd and a dvd.
  - Create a work breakdown structure, based on project deliverables
  - Hints: you will have to write, publish (or record), produce and finally sell, both in retail stores and online, your new project; each of these items may be sold separately

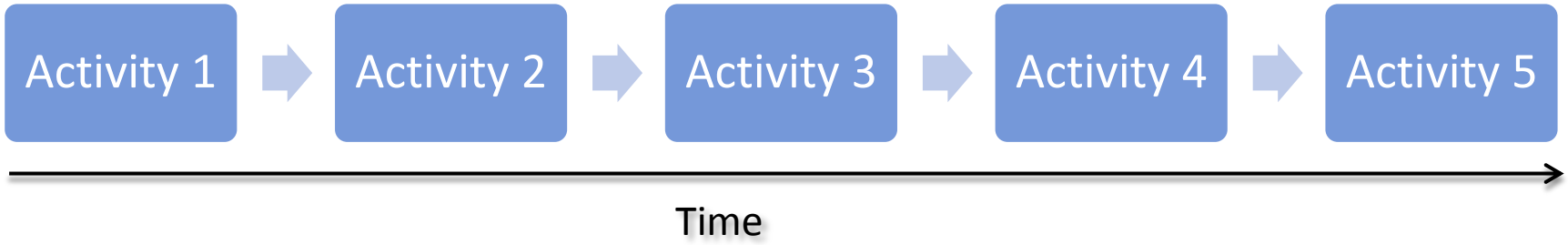
How much detail (finding the sweet spot)

- The 8/80 rule (of thumb)
  - No work package should be less than 8 hours or more than 80 hours
  - Groups of tasks, or activities, once complete, should correspond to the completeness of the corresponding upper level

Scheduling activities

Naïve project scheduling

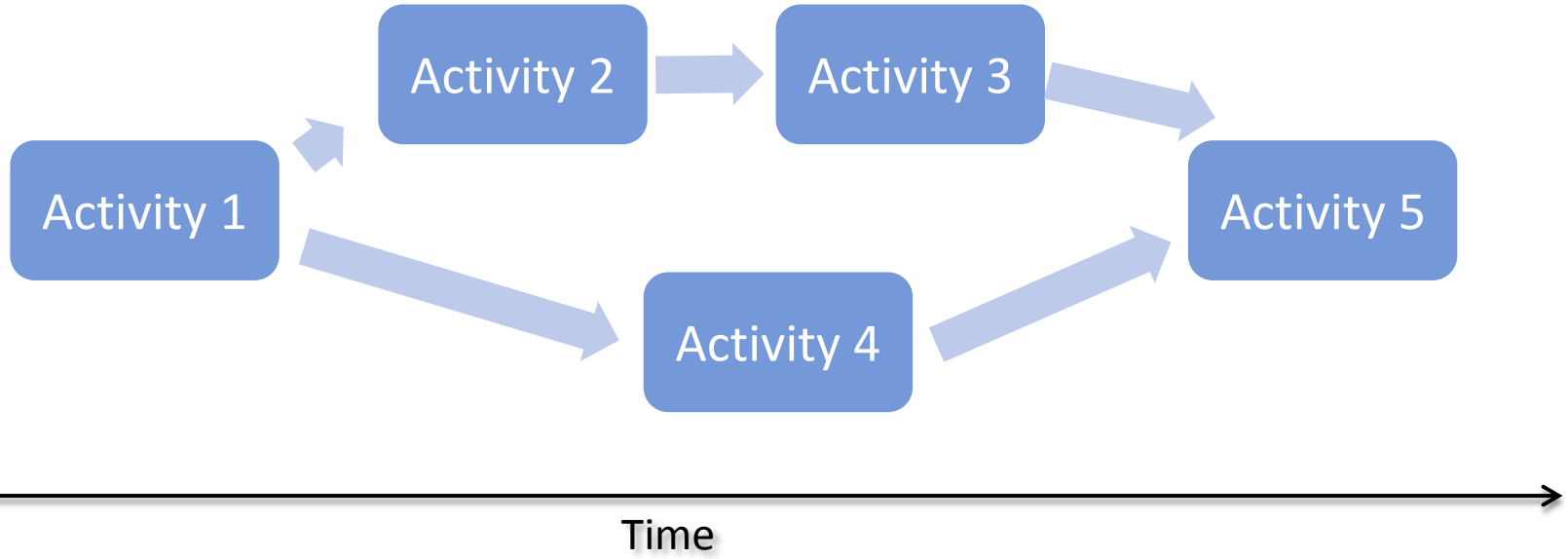
- Project:
  - A sequence of interconnected activities to achieve a certain goal



- Problem:
  - Longest possible completion schedule

Networked project plan

- Build a network of relationships among activities, so that activities precedences can be established
  - Which activities must finish, before a particular new activity starts?

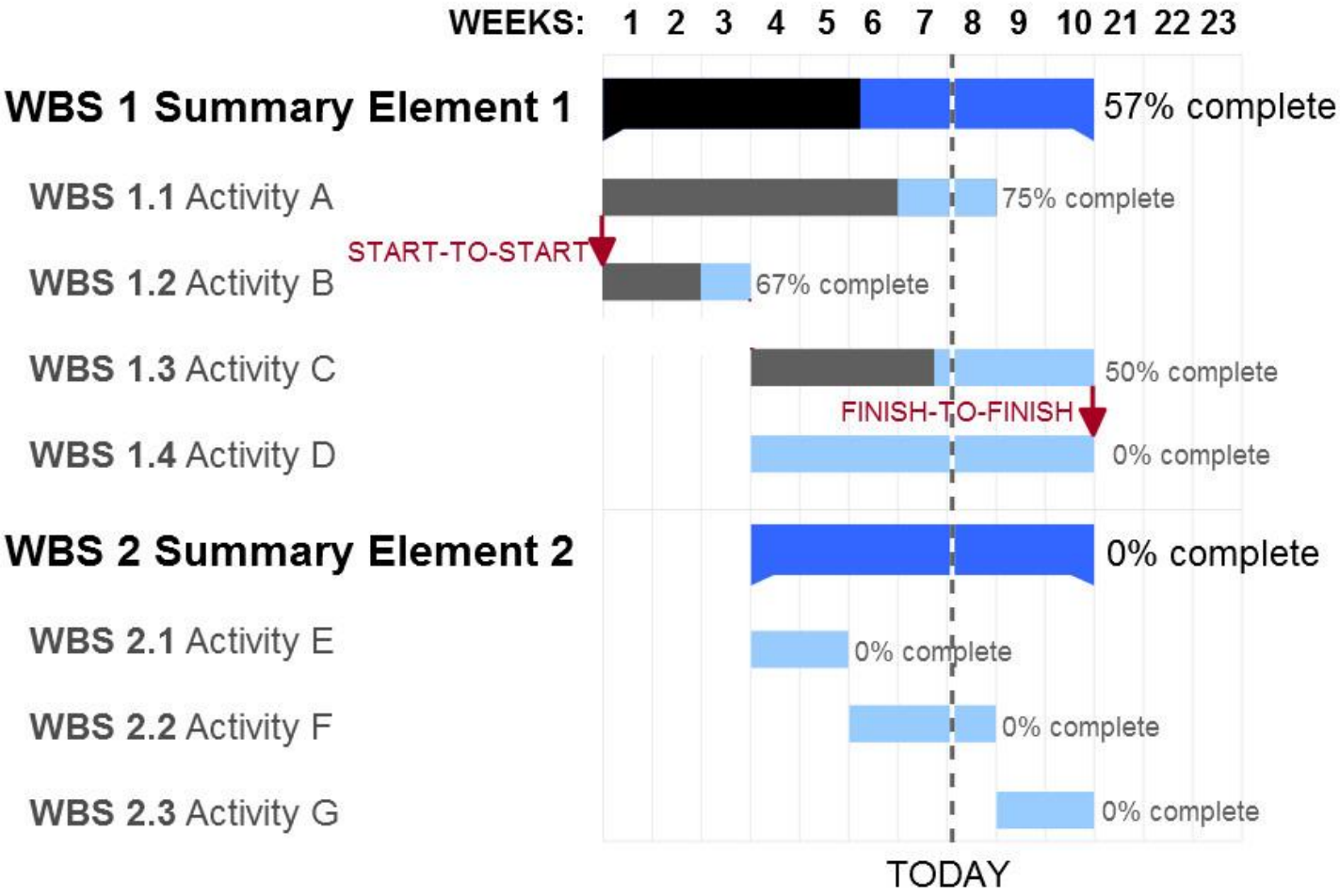


Gantt charts

Gantt charts

- Bar charts to illustrate project schedule
- Work-Breakdown Structure (WBS) with
  - Summary elements
  - Terminal elements
  - Dependencies
- Both Terminal and Summary elements characterized by:
  - Start
  - Finish
  - Current schedule status (completion percentage)
- Visualization of current date
- Well suited for relatively small projects
- Well-known in industry and academia (~100 years old)

Gantt Charts (illustration)



Exercise: Create a Gantt chart for Norman Spaghetti a la Mirjan

- Norman Spaghetti a la Mirjam
  - Prepare spaghetti
    - Cook (9 minutes): spaghetti_0(spaghetti) => spaghetti_1(spaghetti)
    - pour (2 minutes): spaghetti_1(spaghetti) => spaghetti_2(spaghetti)
  - Prepare sauce
    - cut (5 minutes): salmon_0(salmon) => salmon_1(salmon)
    - melt (2 minutes): butter_0(butter) => butter_1(butter)
  - Mix it all
    - add (1 minute): spaghetti_2(spaghetti) butter_1(butter) => mixture_0(spaghetti,butter)
    - mix (1 minute): mixture_0(spaghetti,butter) cream_0(cream) => mixture_1(spaghetti,butter,cream)
    - sprinkle (2minutes): mixture_1(spaghetti,butter,cream) salmon_1(salmon) => mixture_2(spaghetti,butter,cream,salmon)
    - sprinkle (3 minutes): mixture_2(spaghetti,butter,cream, salmon) cheese_0(cheese) => mixture_3(spaghetti,butter,cream,salmon,cheese)

Gantt charts limitations

- Lack of detail on task precedence
- No support for helping the project manager
  - Defining the shortest possible completion schedule
  - Allocating resources effectively

Activity On Arrow diagrams

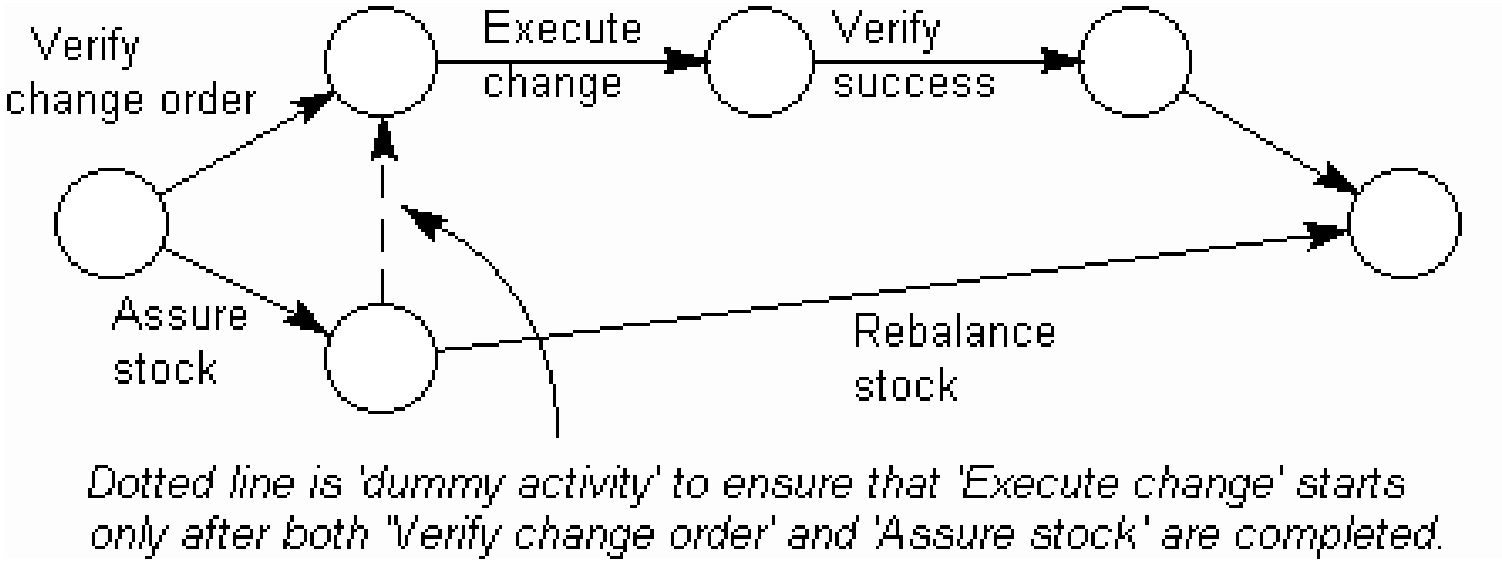
(the most common of these is the Pert diagram)

Pert diagrams

- Detailed project planning & control
- Support for project schedule analytics
  - Identification of the first possible moment for completing an activity
  - Support for earliest completion date computation
  - Comparison of alternative detailed scheduling
  - Project scheduling control

Activity On Arrow (AOA) Diagrams

- Arrows represent activities execution
- Nodes represent the start (or end) of activities
- Dashed arrows represent fictitious ("dummy") activities with null execution time, used for specifying pre-requisite relationships, in order to preserve network integrity
  - The node event only "occurs" after all the inbound activities finish



Representing timing in AOA diagrams

- A – Activity
- T – Time to complete activity (= EFT-EST = LFT–LST)
- EST – Earliest Start Time
- EFT – Earliest Finish Time
- LST – Latest Start Time
- LFT – Latest Finish Time

Algorithm for building AOA diagrams

1. Identify and list all activities
2. Assign each activity a unique id
3. Identify and list the dependencies among activities
4. Design a preliminary network
5. Estimate activities durations
6. Add activities durations to the network
7. Compute early start times
8. Compute late start times
9. Fine tune the network
10. Assign resources

Critical path

- The longest network path formed by activities where EST = LST
  - This is called the **critical path**
    - Any deviation on the duration of activities in this path will have a direct impact on the whole network (i.e. on the whole project schedule)

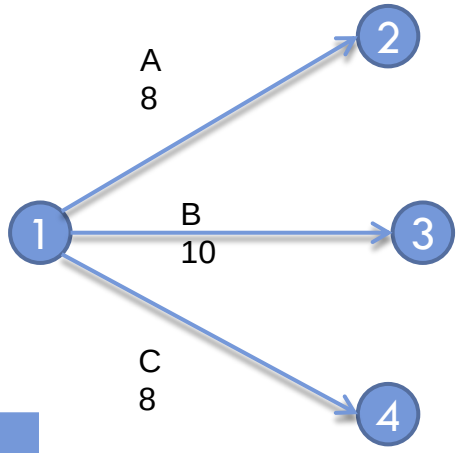
Exercise: create an AOA diagram

Activity	Predecessors	Duration (days)
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	D,B	17
G	C	18
H	C	14
I	F,G	9

Exercise: create an AOA diagram

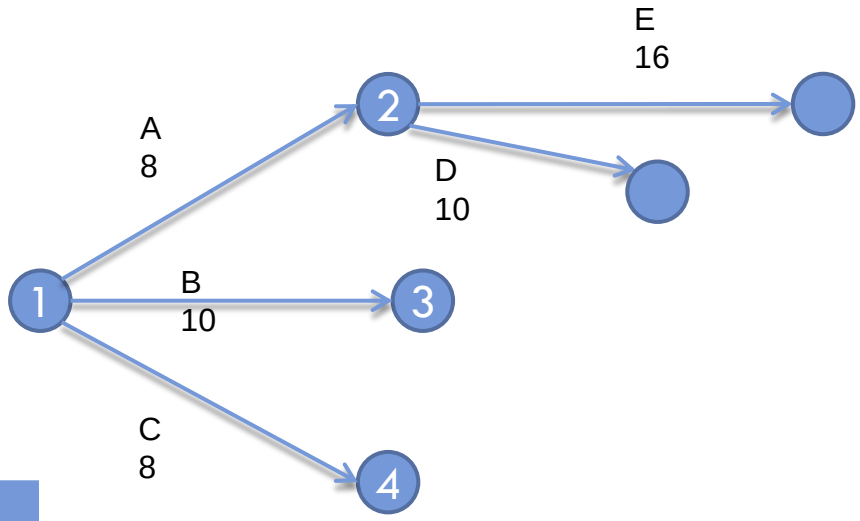
Act	Pred	Dur
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	D,B	17
G	C	18
H	C	14
I	F,G	9

Exercise: create an AOA diagram



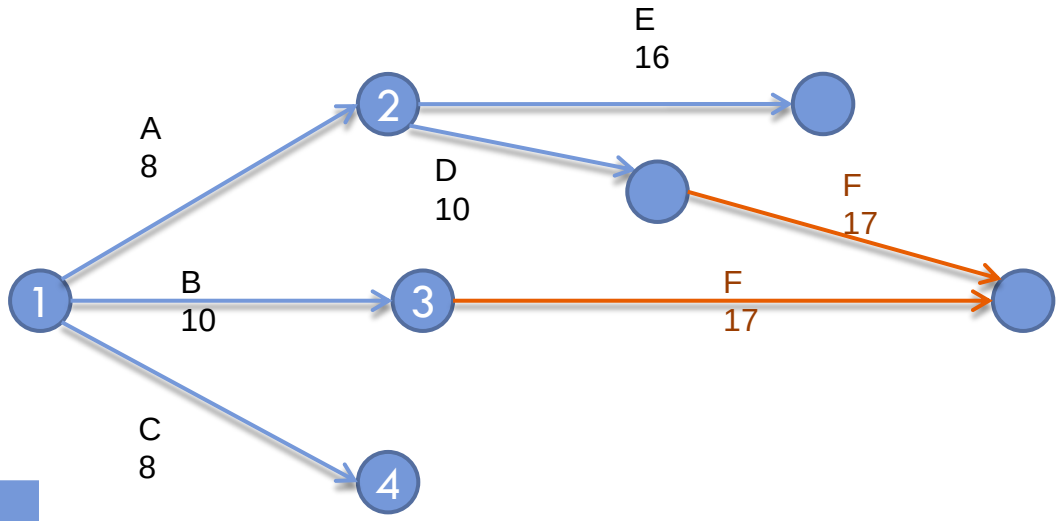
Act	Pred	Dur
A	**-**	**8**
B	**-**	**10**
C	**-**	**8**
D	A	10
E	A	16
F	D,B	17
G	C	18
H	C	14
I	F,G	9

Exercise: create an AOA diagram



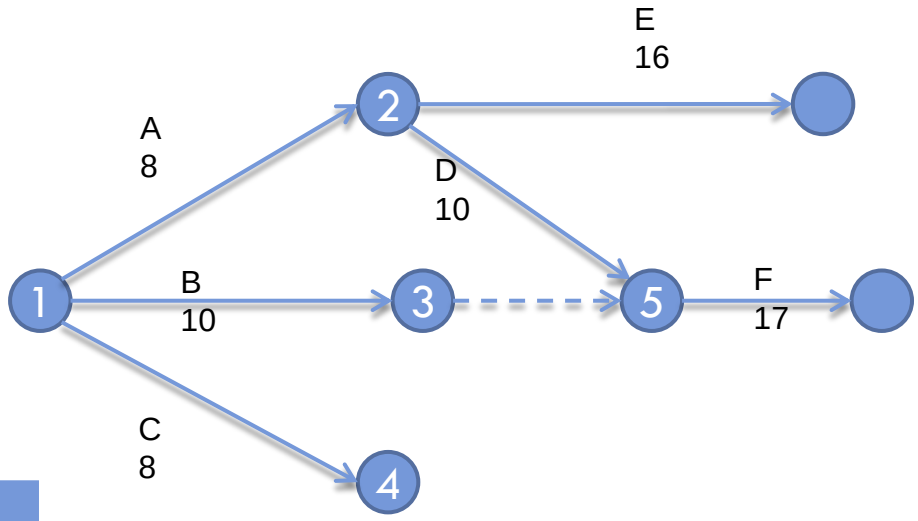
Act	Pred	Dur
A	-	8
B	-	10
C	-	8
D	**A**	**10**
E	**A**	**16**
F	D,B	17
G	C	18
H	C	14
I	F,G	9

Exercise: create an AOA diagram



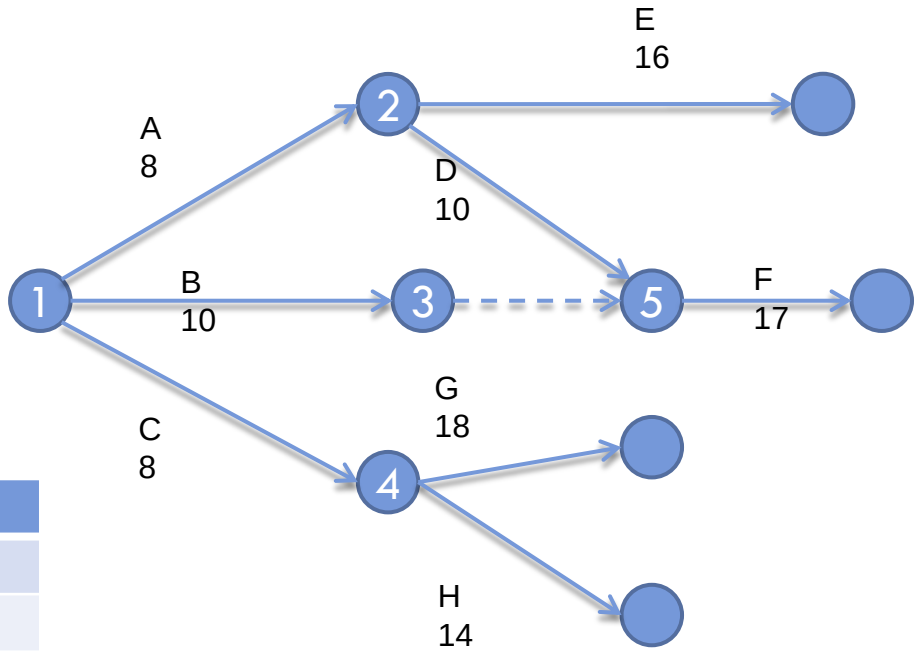
Act	Pred	Dur
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	**D,B**	**17**
G	C	18
H	C	14
I	F,G	9

Exercise: create an AOA diagram



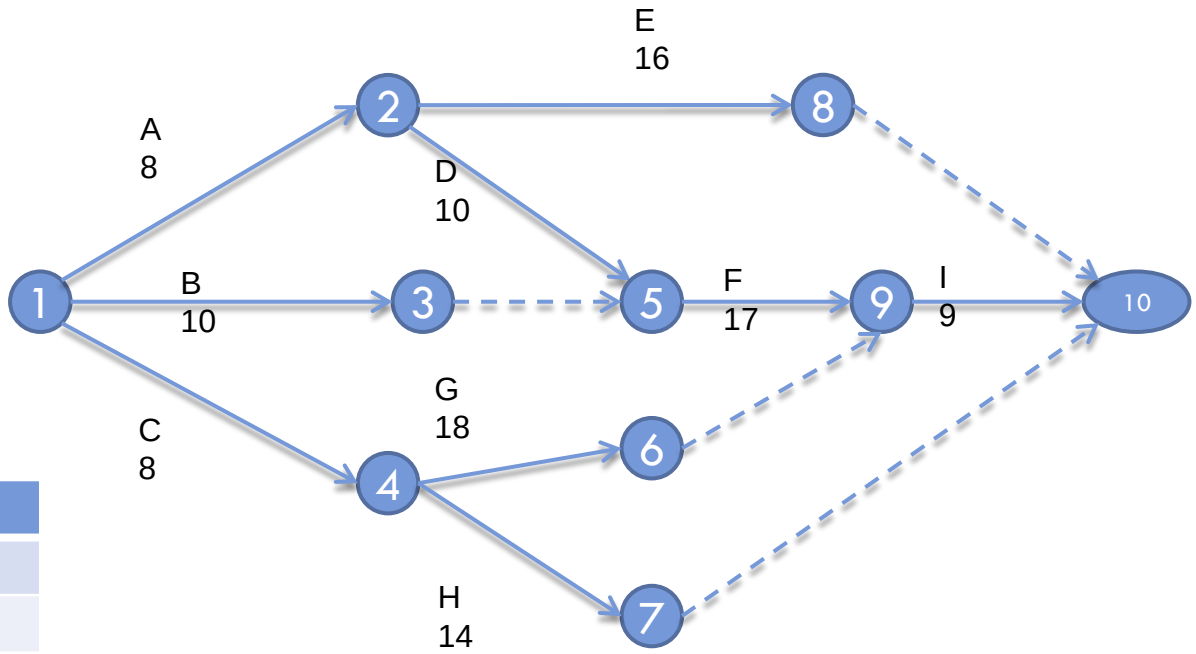
Act	Pred	Dur
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	**D,B**	**17**
G	C	18
H	C	14
I	F,G	9

Exercise: create an AOA diagram



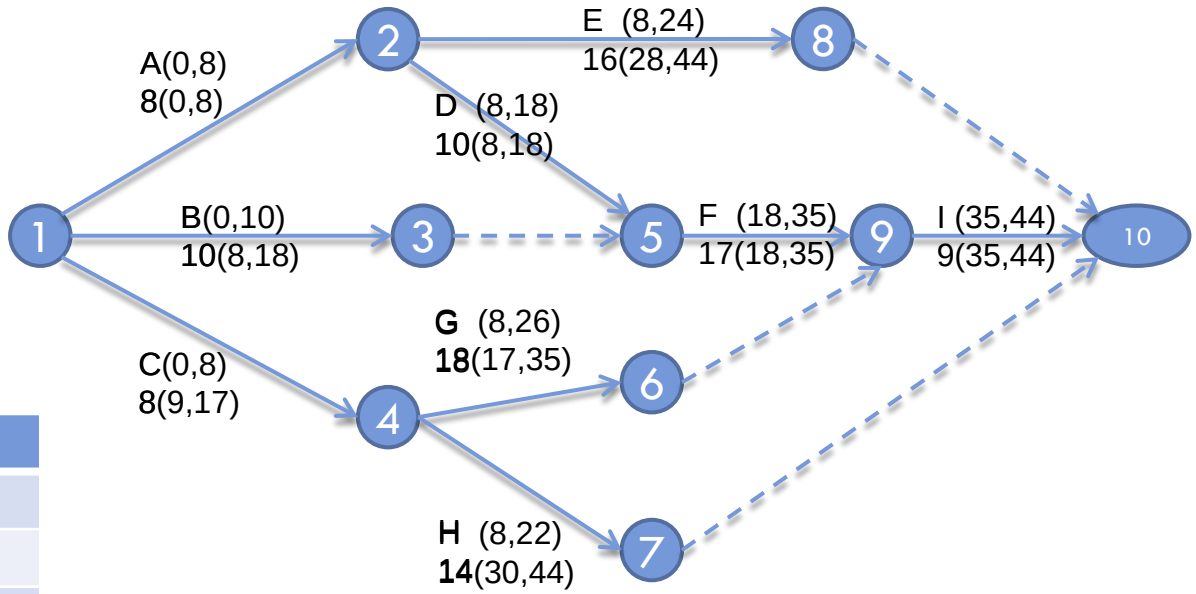
Act	Pred	Dur
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	D,B	17
G	**C**	**18**
H	**C**	**14**
I	F,G	9

Exercise: create an AOA diagram



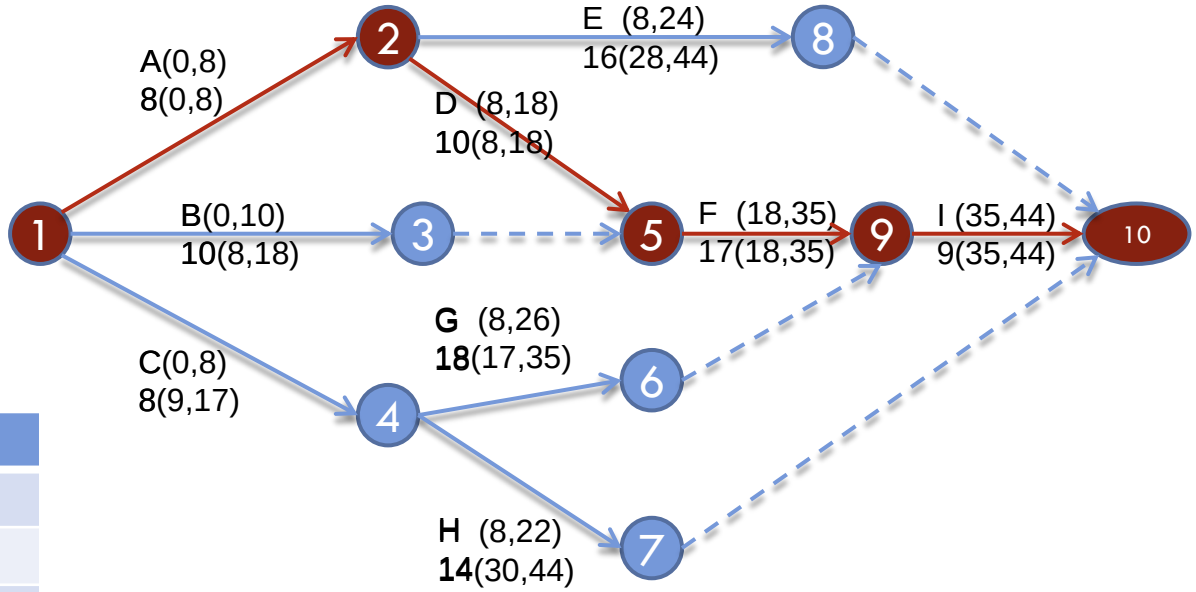
Act	Pred	Dur
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	D,B	17
G	C	18
H	C	14
I	**F,G**	**9**

Exercise: create an AOA diagram



Act	Pred	Dur
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	D,B	17
G	C	18
H	C	14
I	F,G	9

Exercise: create an AOA diagram



Act	Pred	Dur
A	-	8
B	-	10
C	-	8
D	A	10
E	A	16
F	D,B	17
G	C	18
H	C	14
I	F,G	9

A-E -> 8 + 16 = 24

A-D-F-I -> 8 + 10 +17 + 9 = 44

B-F-I -> 10 + 17 + 9 = 36

C-G-I -> 8 + 18 + 9 = 33

C-H-> 8 + 14 = 22

Exercise: create an AOA diagram and compute the critical path

Activity	Precedent Activities	Duration (days)
A	C	40
B	-	27
C	B	32
D	-	57
E	B, D	29
F	H, E	32
G	-	43
H	G	27
I	H, E	39
J	C	41
K	A, J	20

Limitations of AOA diagrams

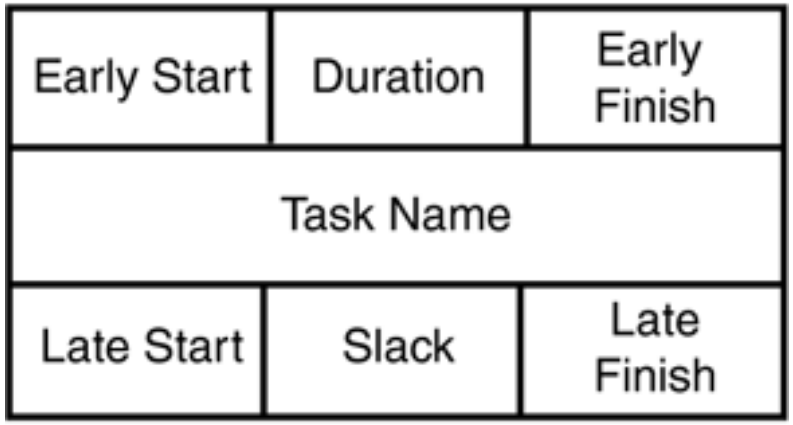
- The need to create “dummy” activities for preserving network integrity
  - Only simple dependences can be represented
  - Accidental complexity makes this technique hard to apply when projects include many activities

Activity On Node Diagrams

(the Precendence Diagram Method)

Precedence Diagram Method

- Each activity represented by an **activity node**



Duration = Early Finish – Early Start = Late Start – Late Finish
Slack = Late Start – Early Start = Late Finish – Early Finish

- **Predecessor/successor (aka dependence)** relationships represented by arrows

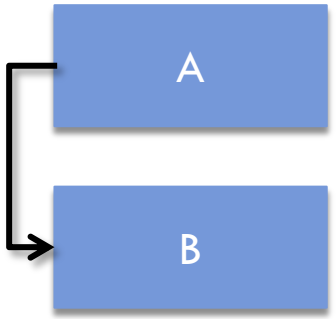
Dependence relationships (FS)

- **F**inish to **S**tart relationship
  - As soon as A finishes, B can start

- Example
  - A is a data collection activity
  - B is a data storing activity
  - As soon as we finish data collection, we can start data storing
- Notes:
  - Recommended as default dependency in early planning

Dependence relationships (SS)

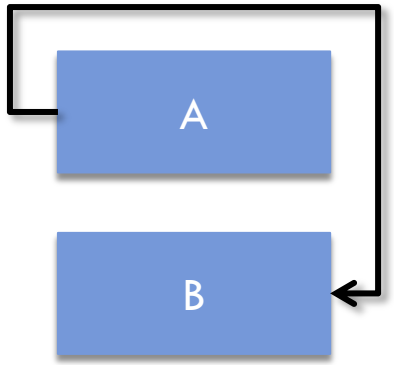
- **S**tart to **S**tart relationship
  - As soon as A starts, B can start



- Example
  - A is a data collection activity
  - B is a data storing activity
  - Data storing cannot start before data collection starts
- Notes:
  - Use for compressing activities

Dependence relationships (SF)

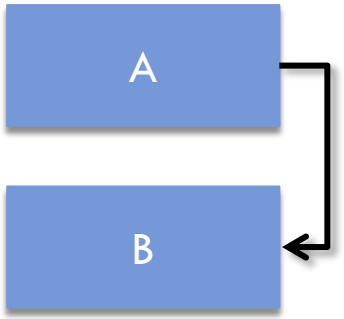
- **S**tart to **F**inish relationship
  - As soon as A starts, B can finish



- Example
  - A is a new system running
  - B is an old system running
  - As soon as the new system is running, the old system may be discontinued
- Notes:
  - Use for just in time scheduling (relatively uncommon)

Dependence relationships (FF)

- **F**inish to **F**inish relationship
  - As soon as A finishes, B can finish



- Example
  - A is a data collection activity
  - B is a data storing activity
  - Data storing cannot finish before data collection finishes
- Notes:
  - To preserve connectivity in the network, **SS** should be accompanied with **FF**

Creating Activity On Node diagrams

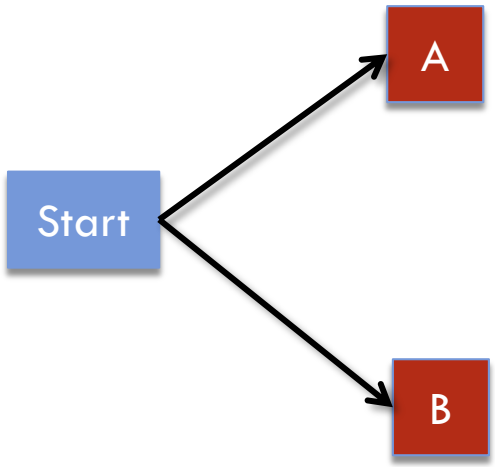
Activity	Predecessors
A	-
B	-
C	A
D	B
E	B
F	A
G	C
H	D
I	A
J	E, G, H
K	F, I, J

How would you build an AON diagram for this process?

Activity on Node diagram

Act	Pred
A	-
B	-
C	A
D	B
E	B
F	A
G	C
H	D
I	A
J	E, G, H
K	F, I, J

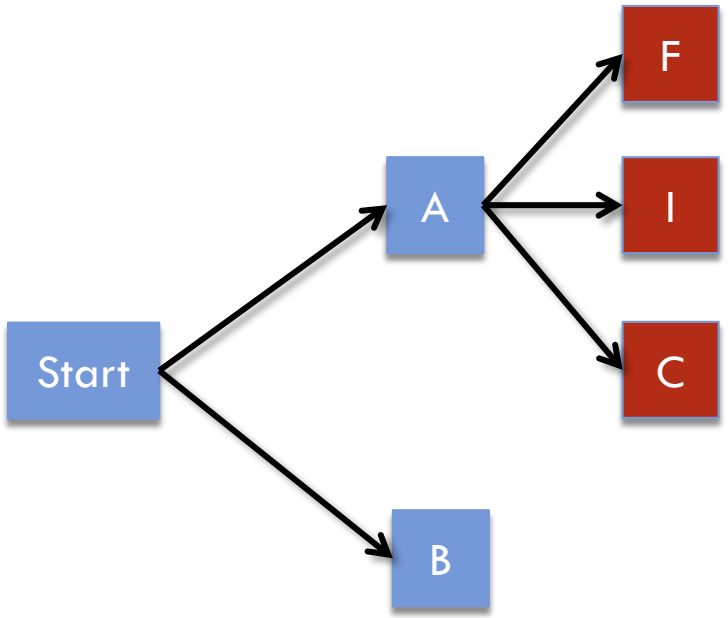
Activity on Node diagram



Act	Pred
A	-
B	-
C	A
D	B
E	B
F	A
G	C
H	D
I	A
J	E, G, H
K	F, I, J

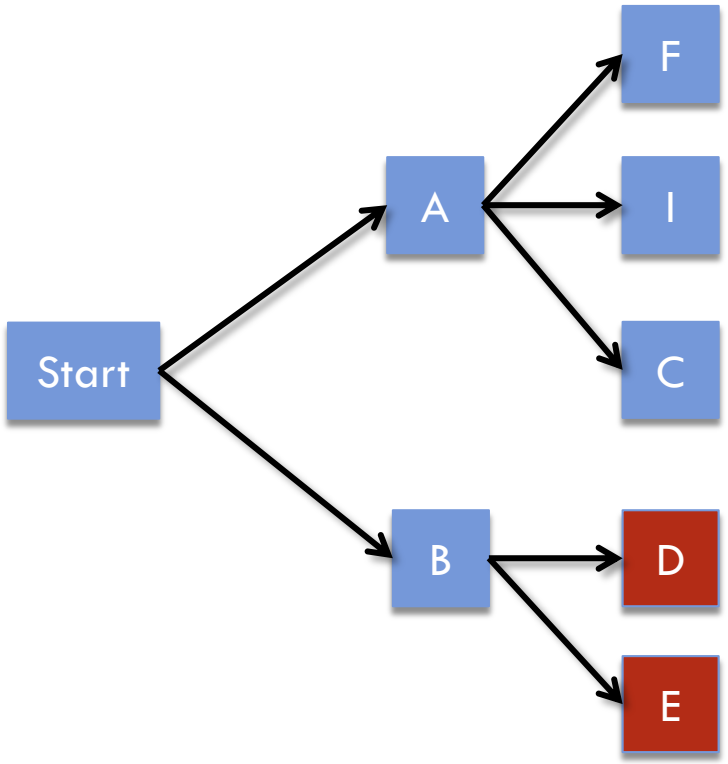
Activity on Node diagram

Note: A is called a **burst** activity: more than one activity can start once A is finished



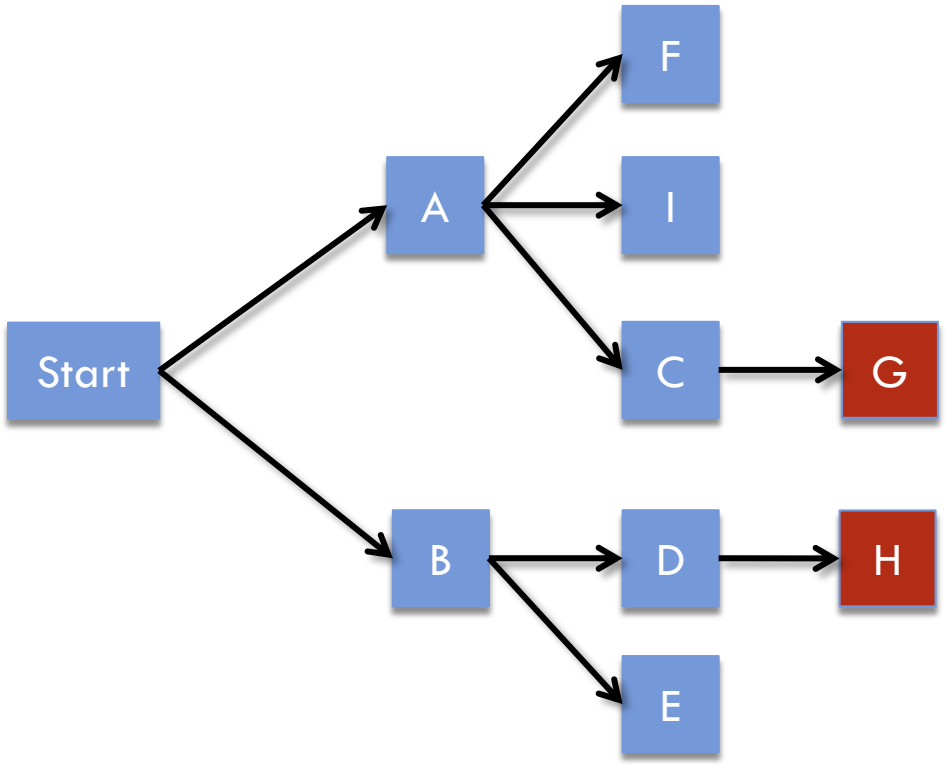
Act	Pred
A	-
B	-
C	A
D	B
E	B
F	A
G	C
H	D
I	A
J	E, G, H
K	F, I, J

Activity on Node diagram



Act	Pred
A	-
B	-
C	A
D	B
E	B
F	A
G	C
H	D
I	A
J	E, G, H
K	F, I, J

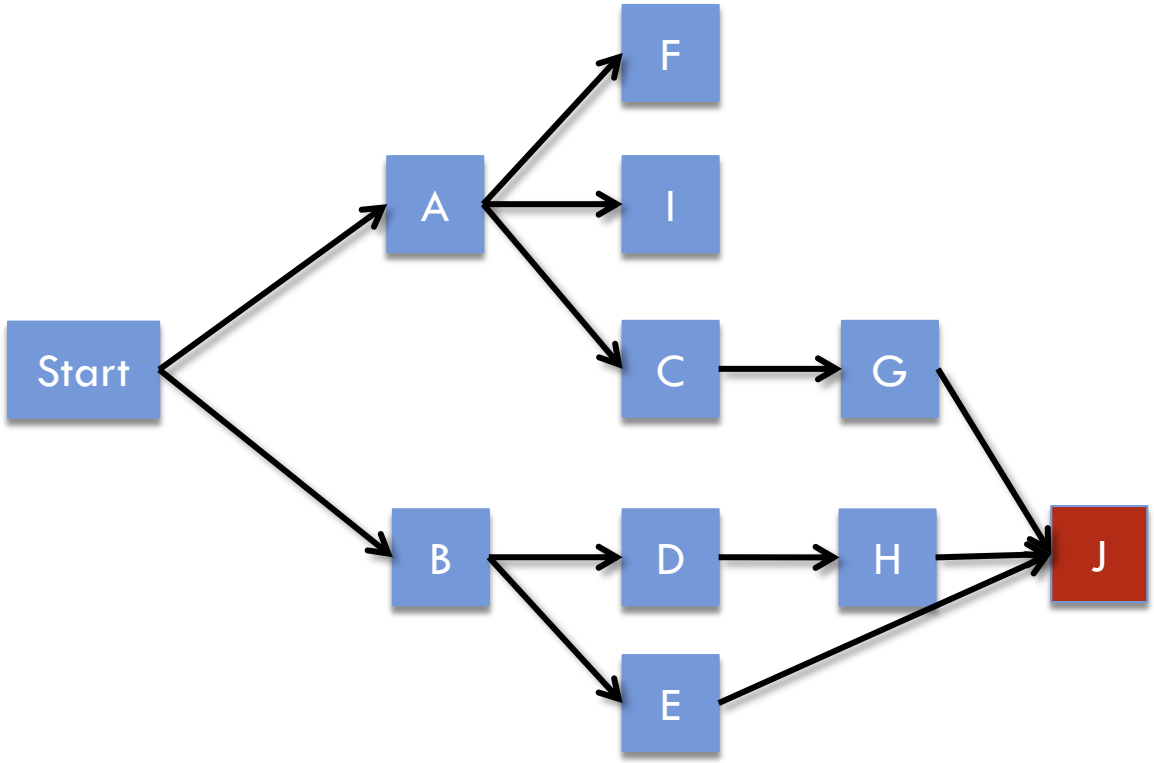
Activity on Node diagram



Act	Pred
A	-
B	-
C	A
D	B
E	B
F	A
G	**C**
H	**D**
I	A
J	E, G, H
K	F, I, J

Activity on Node diagram

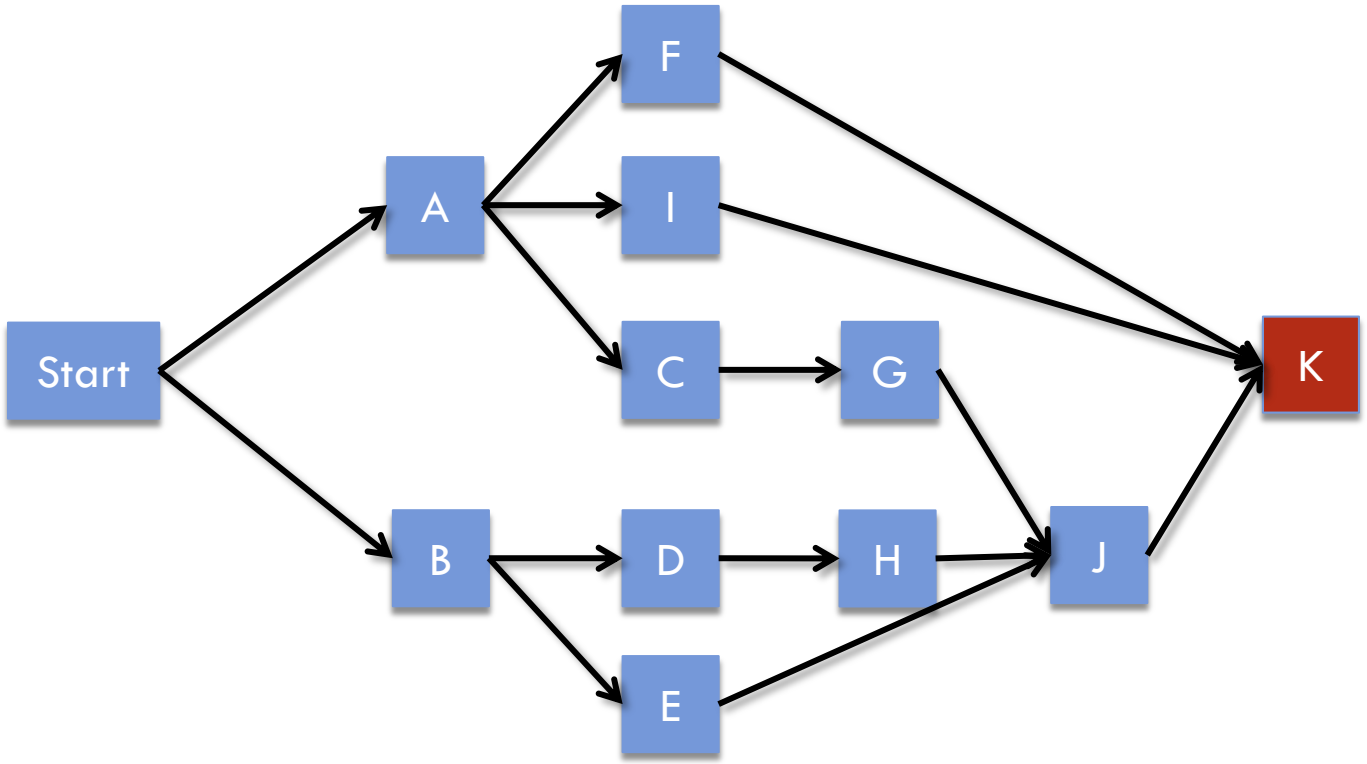
Note: J is called a **merge** activity: J can not start before all its preceding activities are finished.



Act	Pred
A	-
B	-
C	A
D	B
E	B
F	A
G	C
H	D
I	A
J	E, G, H
K	F, I, J

Activity on Node diagram

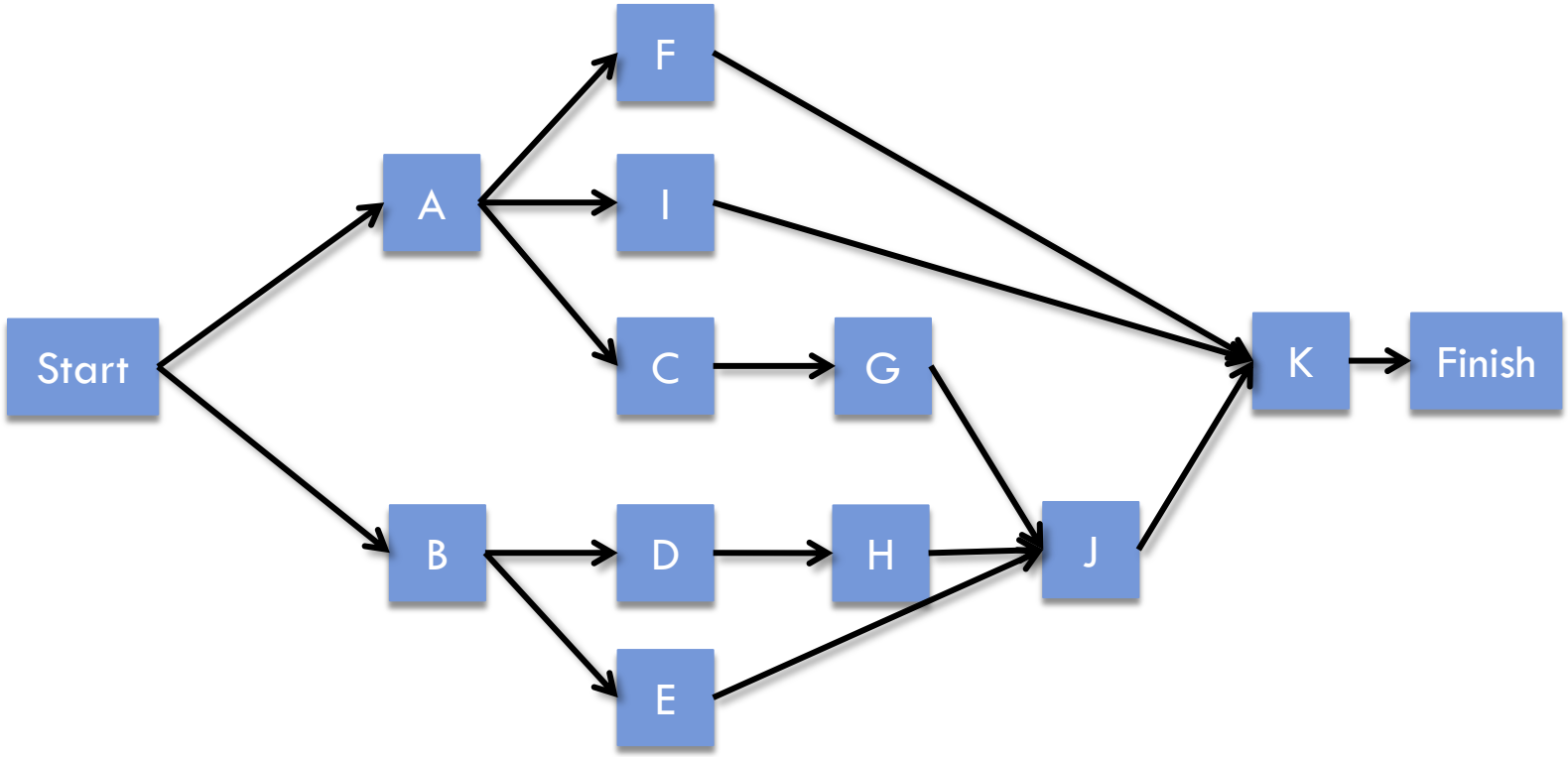
Note: K is called a **merge** activity: K can not start before F, I and J are finished.



Act	Pred
A	-
B	-
C	A
D	B
E	B
F	A
G	C
H	D
I	A
J	E, G, H
K	**F, I, J**

Activity on Node diagram

Act	Pred
A	-
B	-
C	A
D	B
E	B
F	A
G	C
H	D
I	A
J	E, G, H
K	F, I, J



Example:

- A company decides to reengineer its IT system.
- They will need new hardware, network and internet access, along with the corresponding software
- The Project Manager has already made a description of the activities to conduct and the time required for each of them

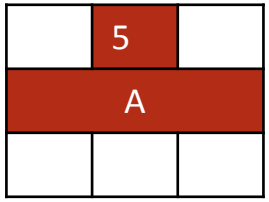
- How much time do we need for this project?

How much time do we need for this project?

Activity	Predecessors	Duration (in months)
A. Plan project	-	5
B. Acquire hardware	A	10
C. Select location	A	6
D. Determine environmental needs - Power - Phones - Workstations (pcs/laptops) - Software - CASE tools - Furniture	A	7
E. Install hardware	B, C	4
F. Install environmental needs	D, E	2
G. Start production	F	2

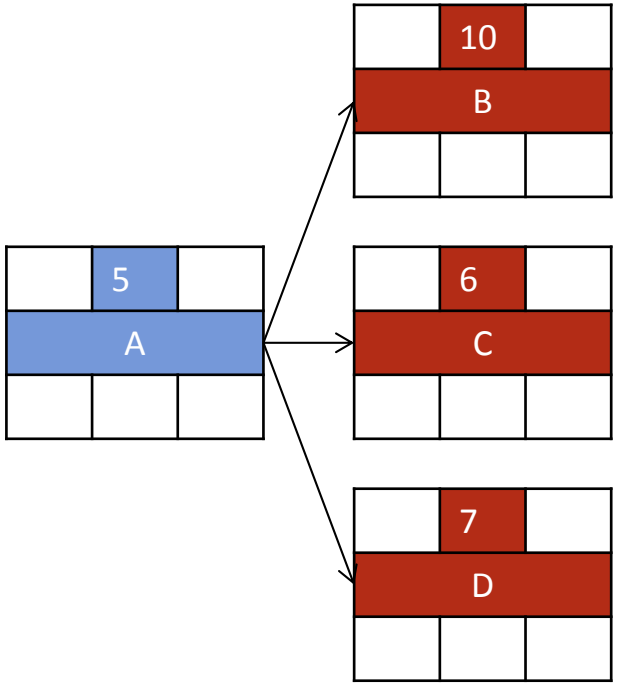
Activity **A** has no predecessors and lasts **5** days

Act	Pred	Dur
A	-	**5**
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



B, C, and D follow A

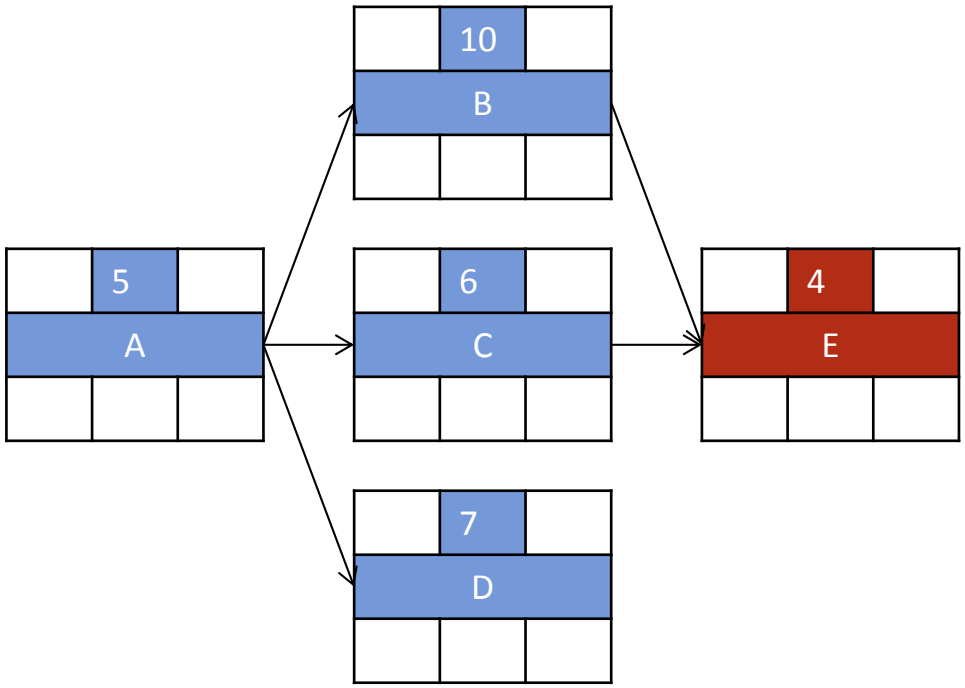
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

E can start after both B and C finish

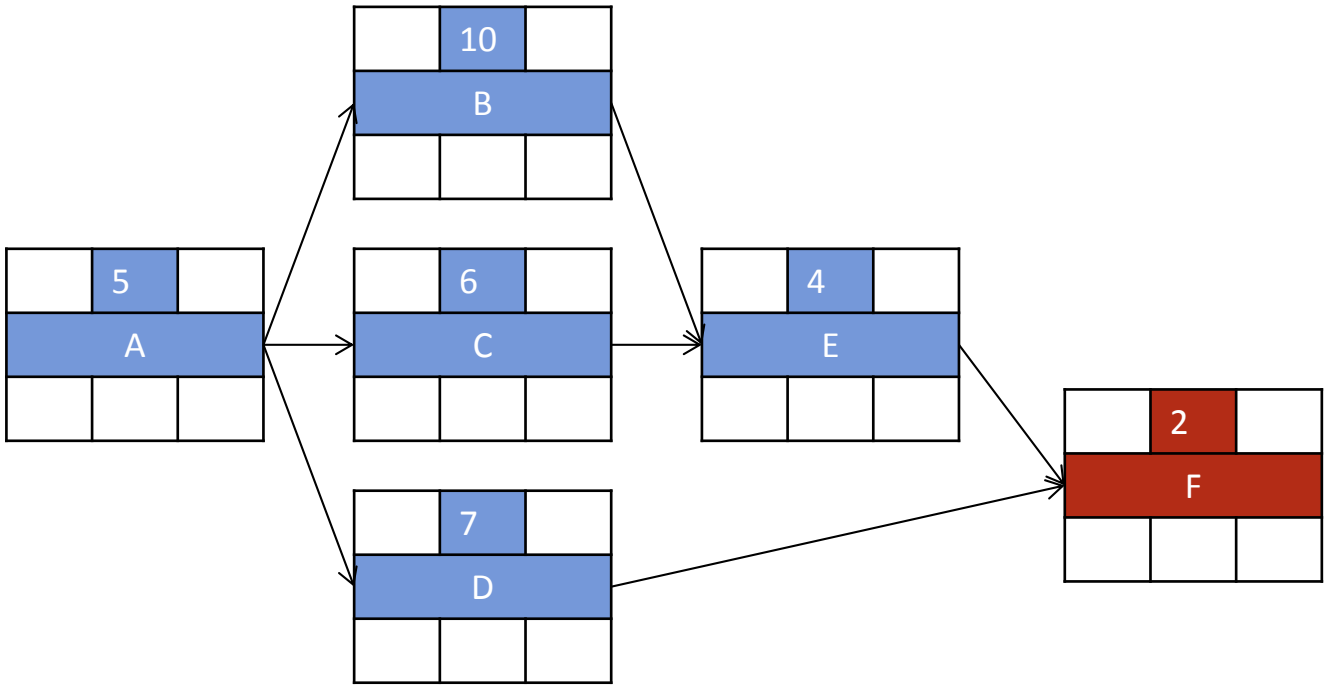
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	**B, C**	**4**
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

F can start after both D and E finish

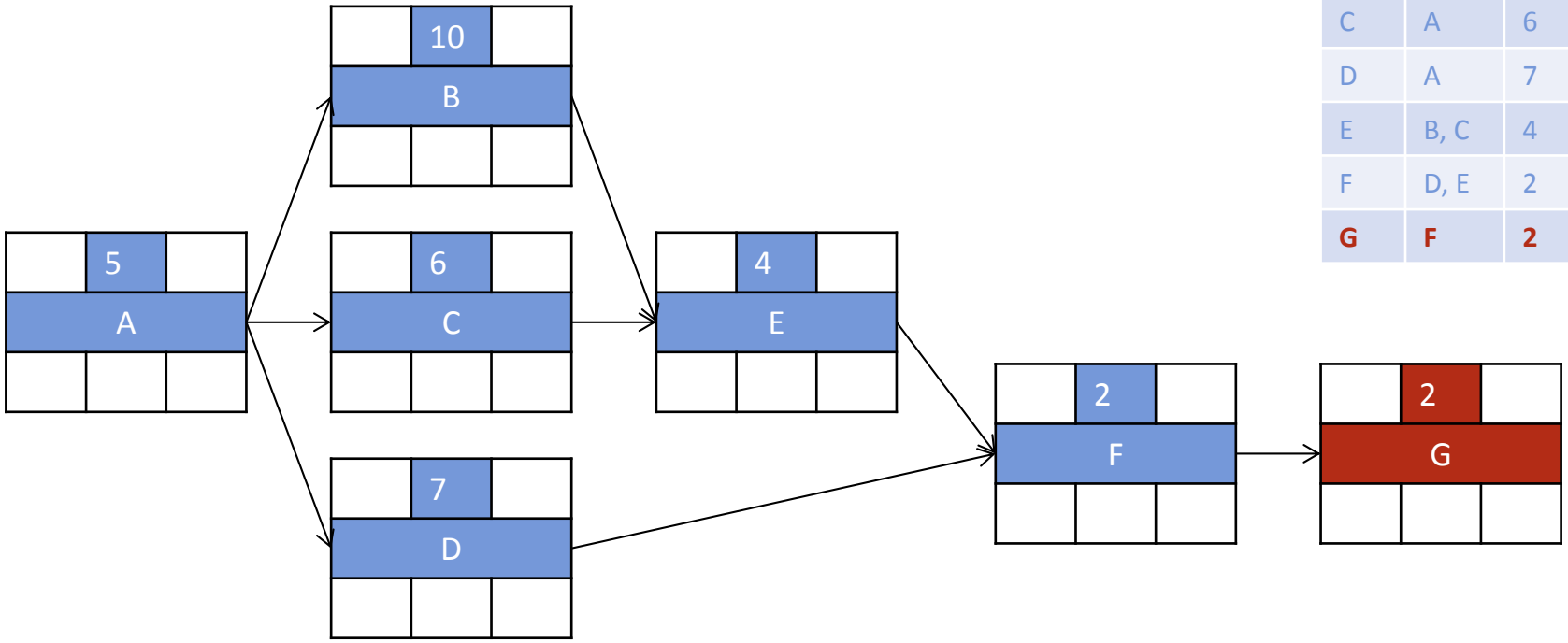
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	**D, E**	**2**
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

G can start after F finishes

Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	**F**	**2**

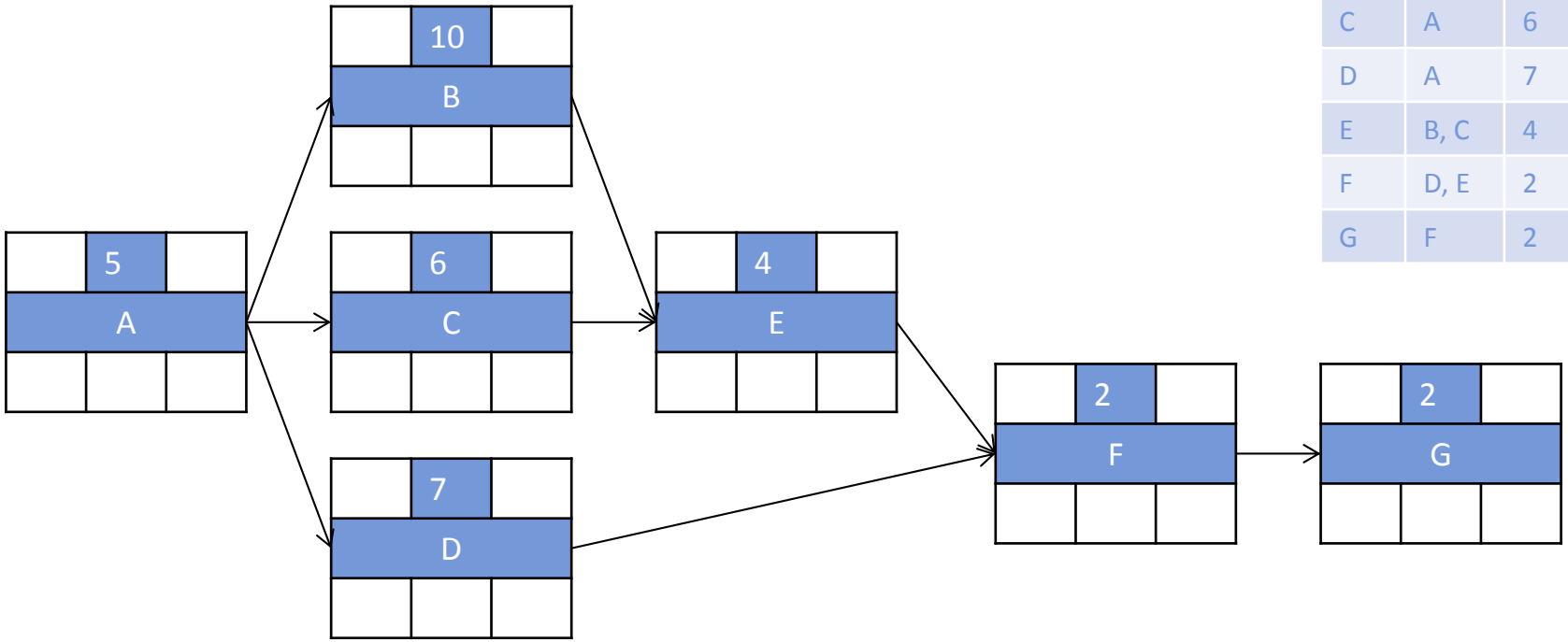


Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

EST(A) = ?
EFT(A) = ?

Note: A is the first activity!
So, EST(A) should be 0

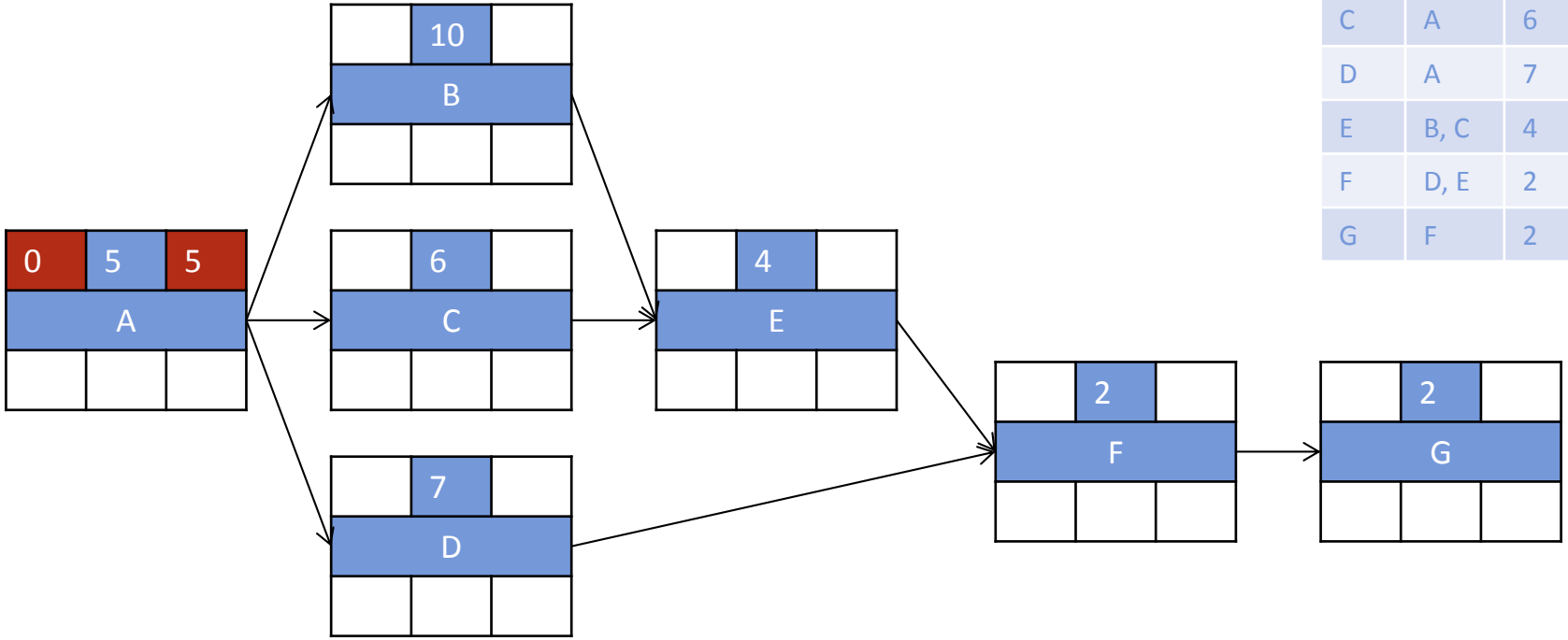
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

EST(A) ← 0
EFT(A) ← EST(A) + DUR(A)

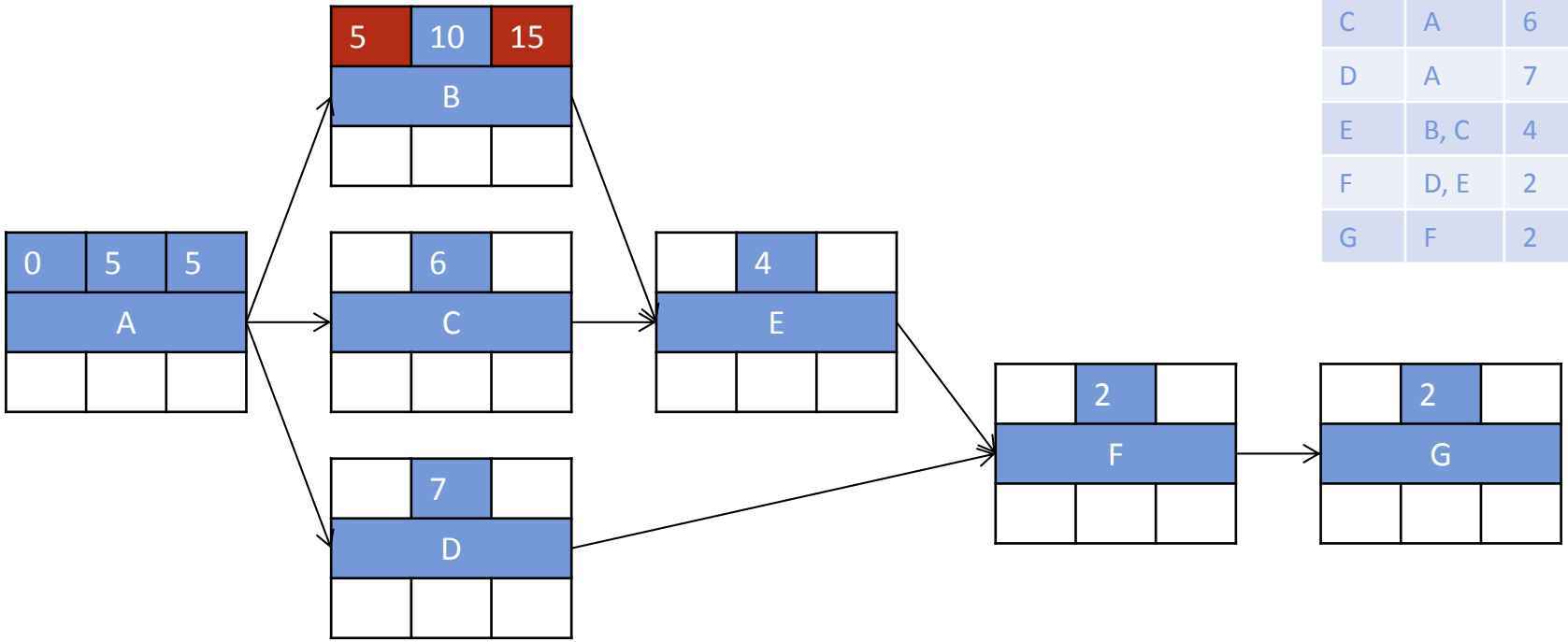
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

EST(B) ← EFT(A)
EFT(B) ← EST(B) + DUR (B)

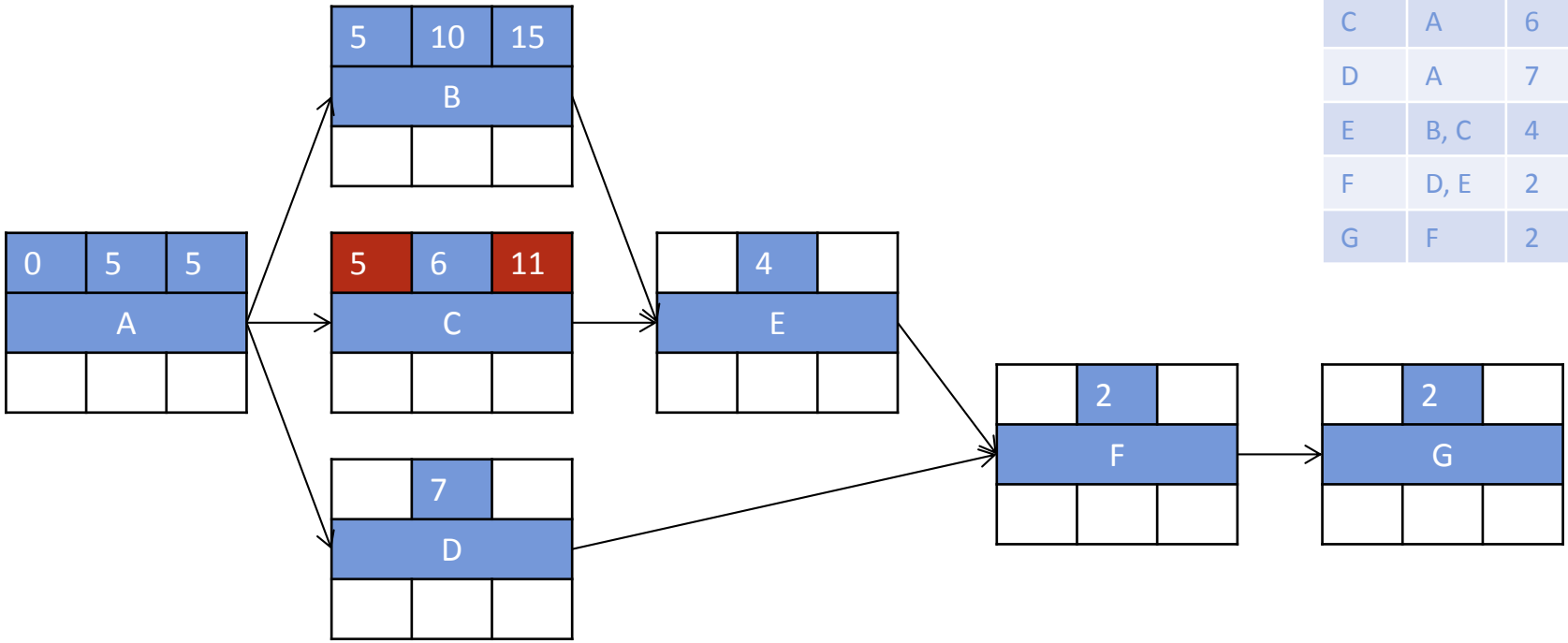
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

EST(C) ← EFT(A)
EFT(C) ← EST(C) + DUR(C)

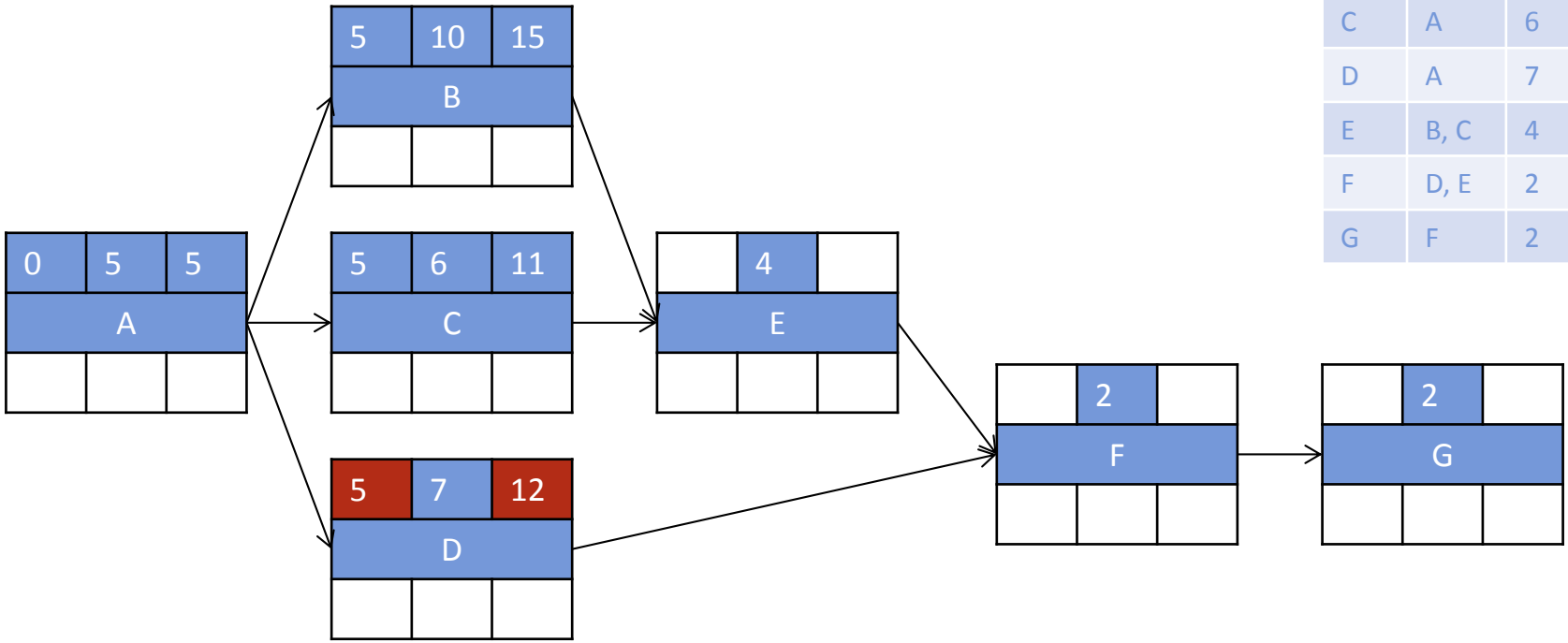
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

EST(D) ← EFT(A)
EFT(D) ← EST(D) + DUR(D)

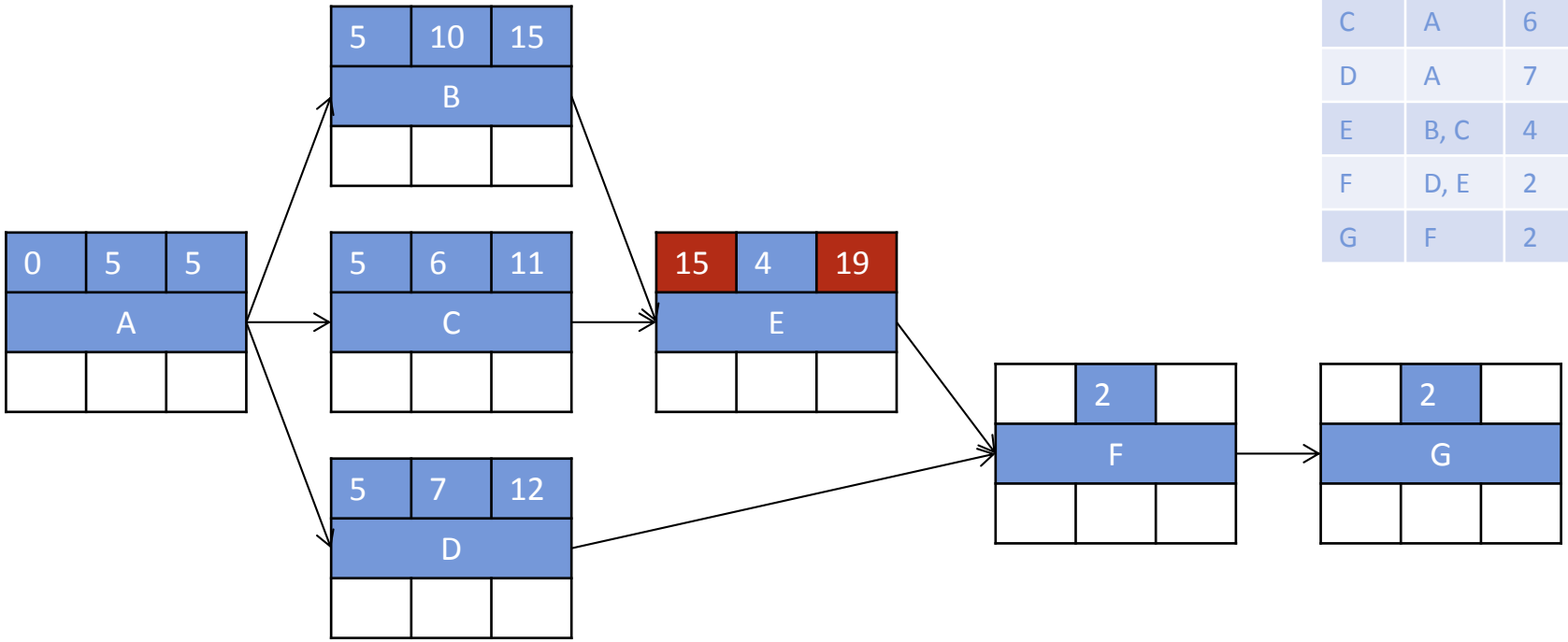
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

EST(E) ← MAX (EFT(B), EFT(C))
EFT(E) ← EST(E) + DUR(E)

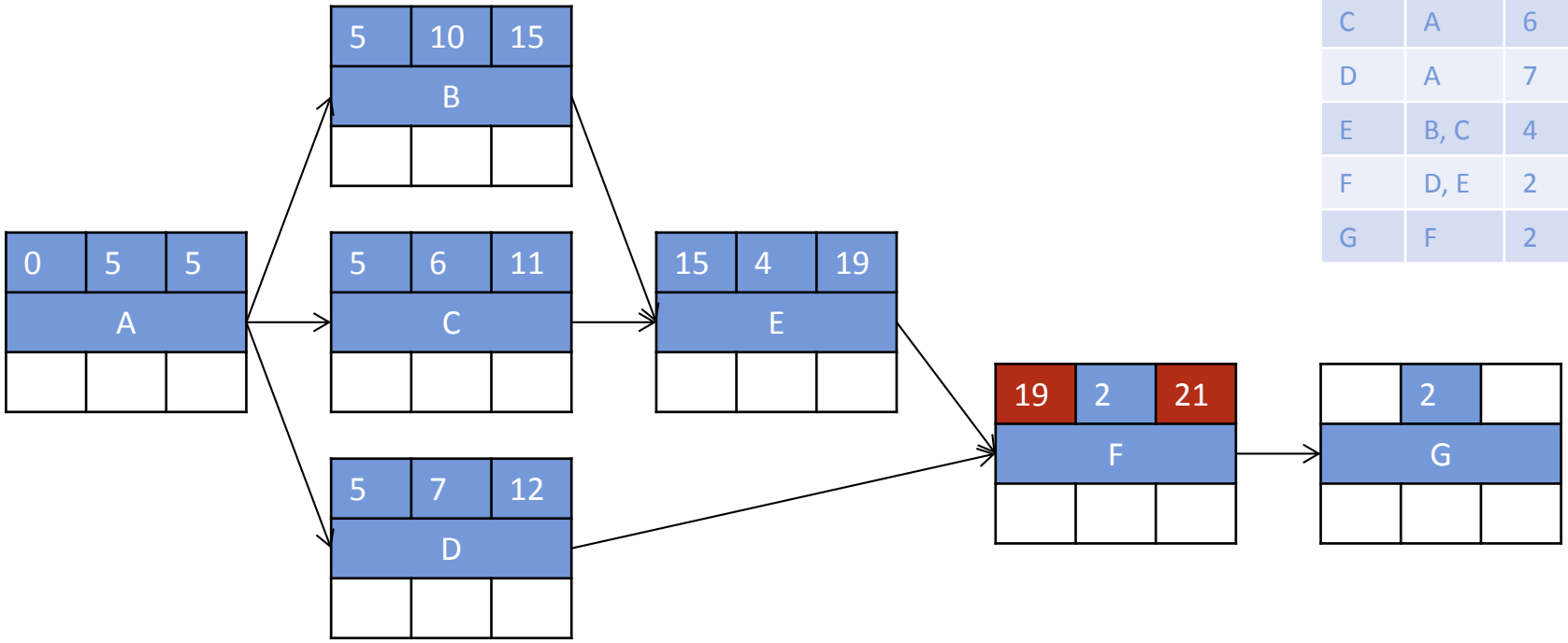
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

EST(F) ← MAX (EFT(D), EFT(E))
EFT(F) ← EST(F) + DUR(F)

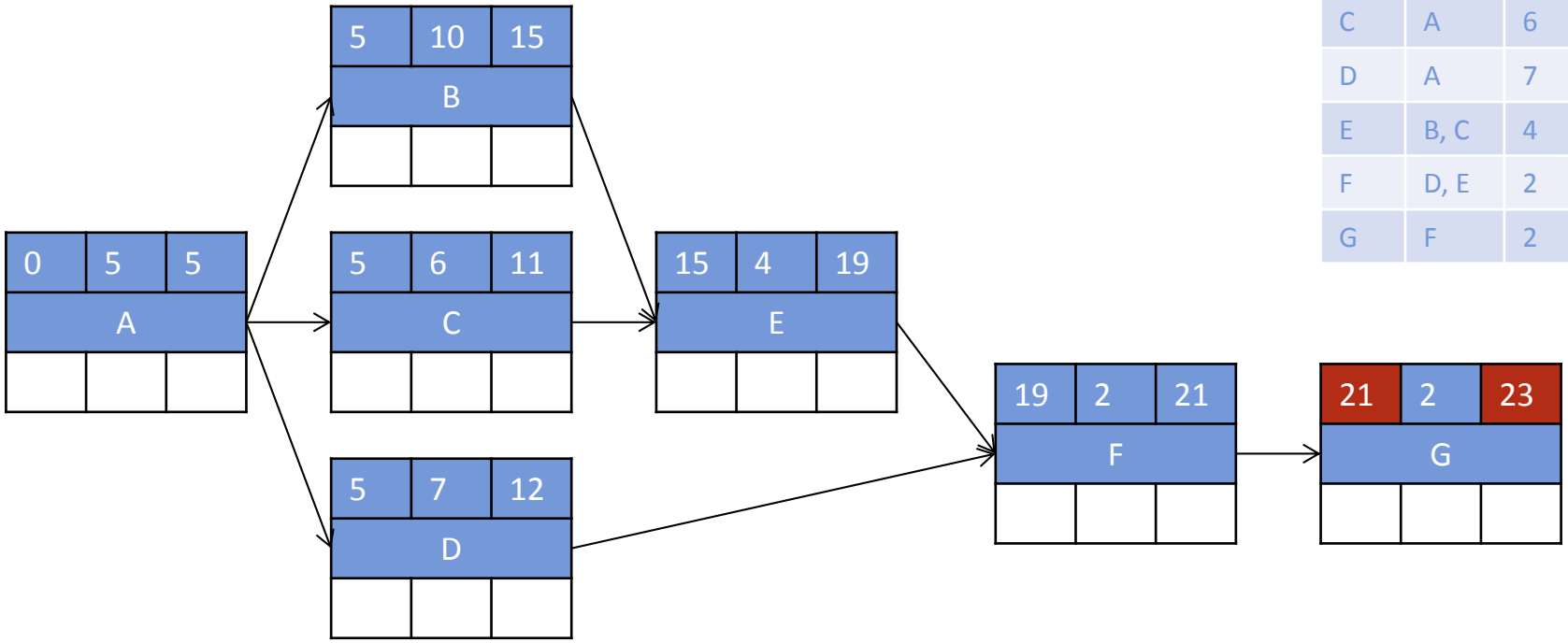
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

EST(G) ← EFT(F)
EFT(G) ← EST(G) + DUR (G)

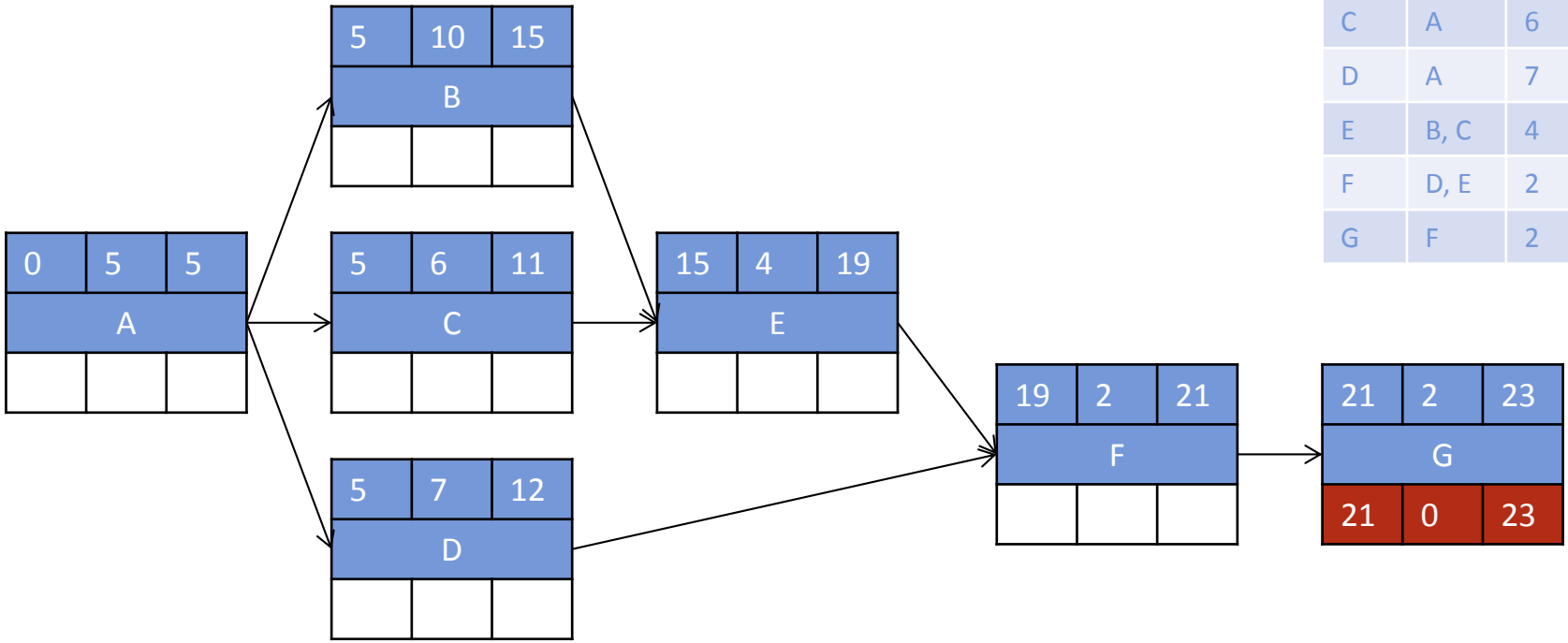
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

LFT(G) ← EFT(G)
LST(G) ← LFT(G) – DUR(G)
SLACK(G) ← LST(G) – EST(G)

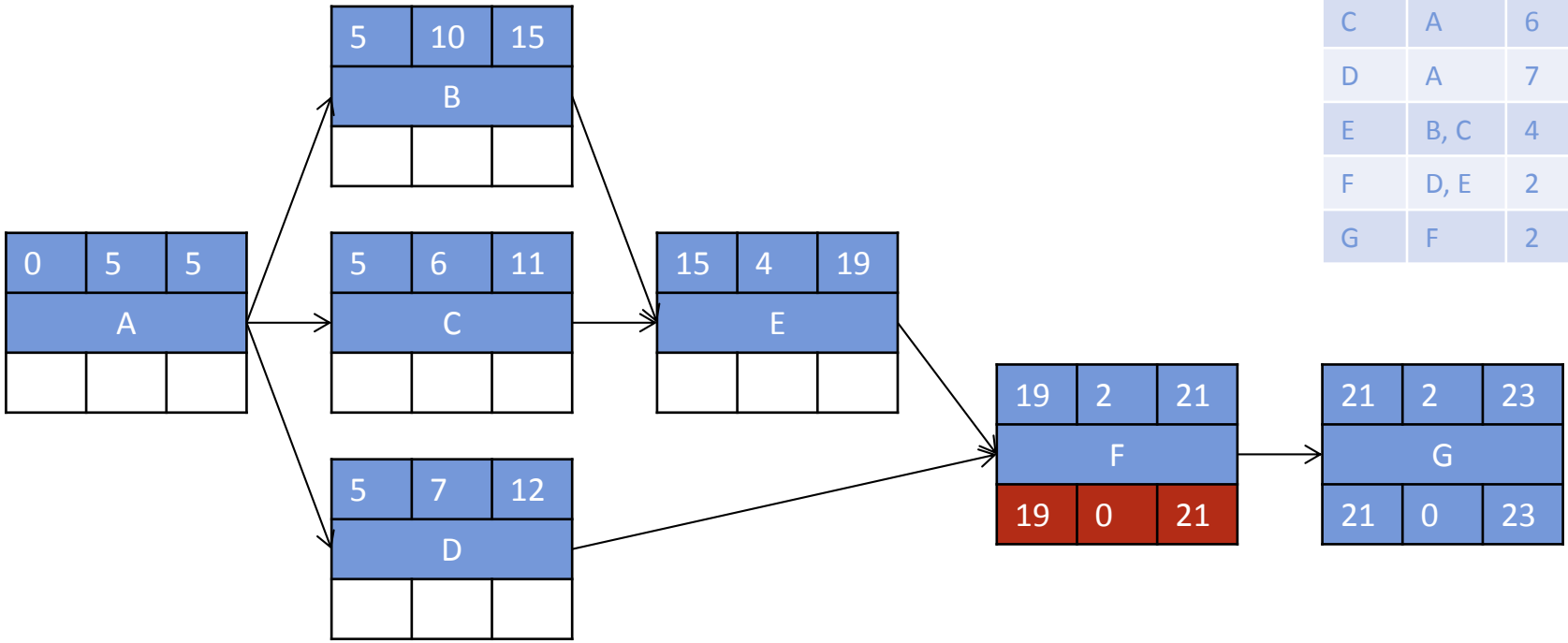
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

LFT(F) ← LST(G)
LST(F) ← LFT(F) – DUR(F)
SLACK(F) ← LST(F) – EST(F)

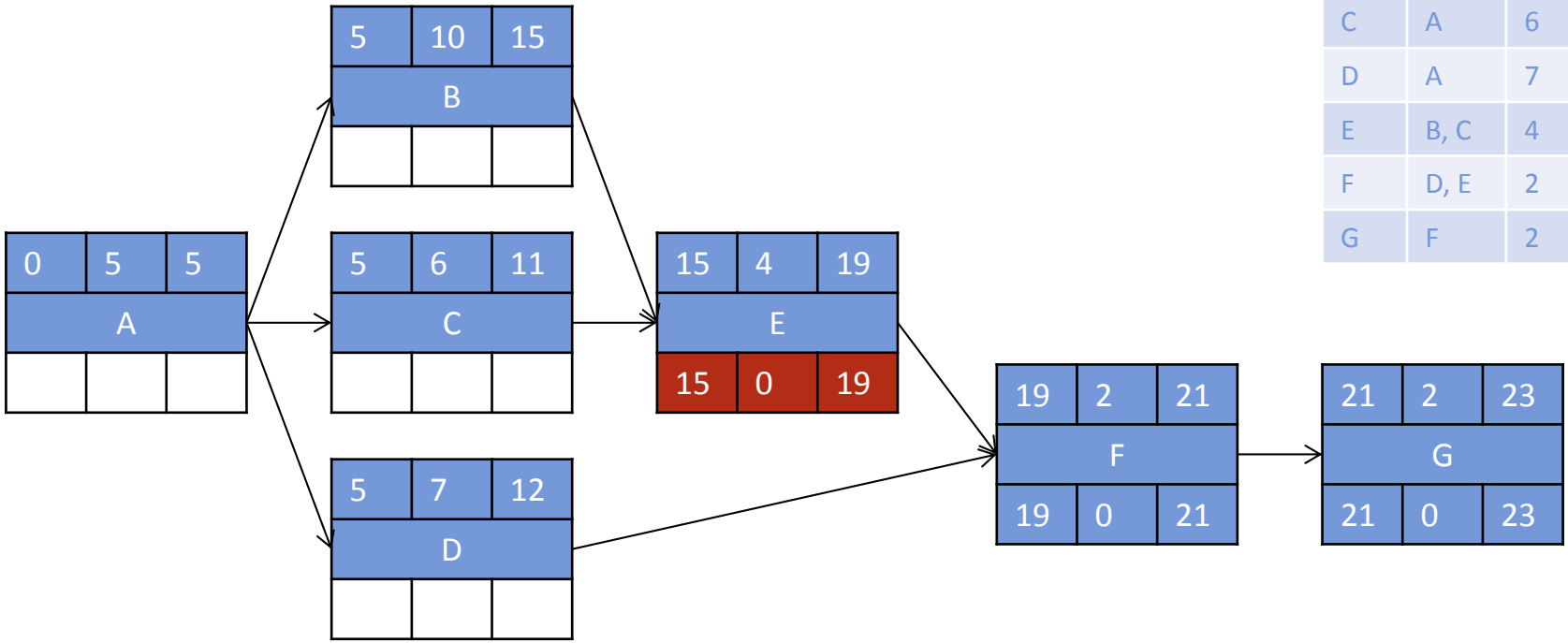
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

LFT(E) ← LST(F)
LST(E) ← LFT(E) – DUR(E)
SLACK(E) ← LST(E) – EST(E)

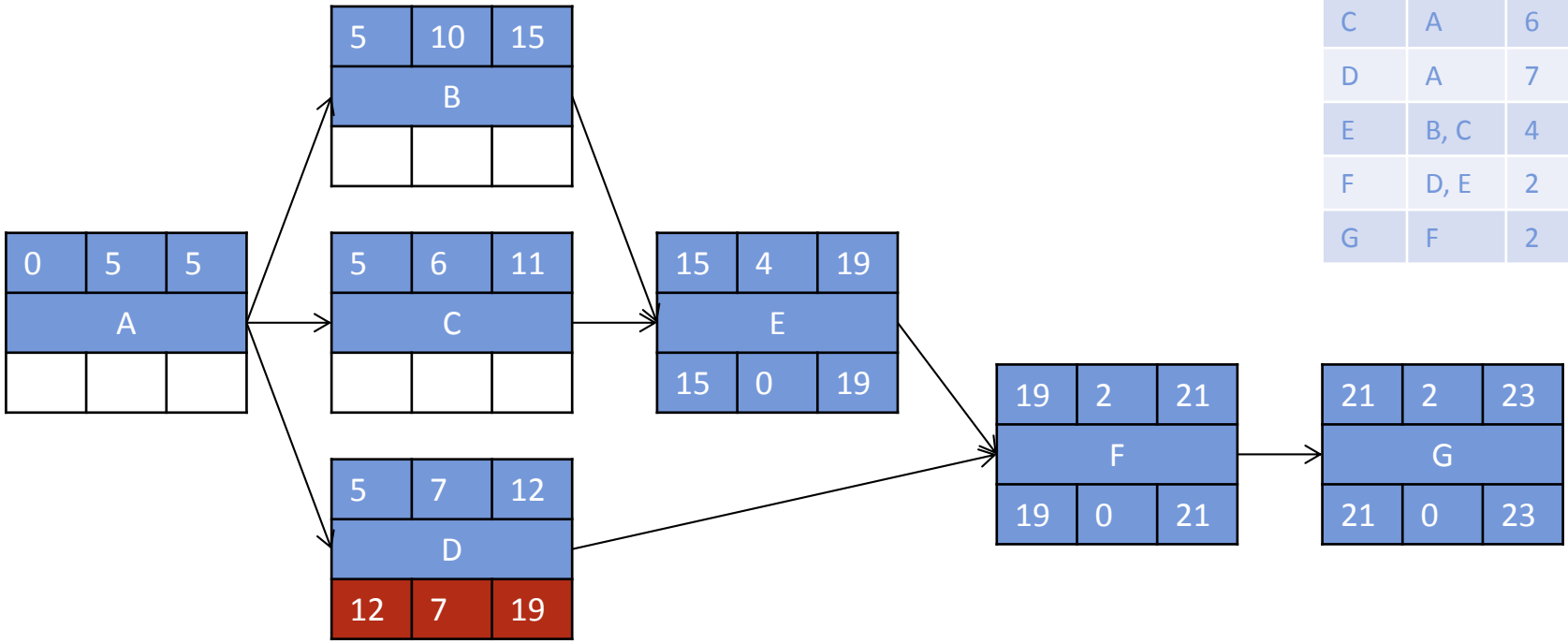
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

LFT(D) ← LST(F)
LST(D) ← LFT(D) – DUR(D)
SLACK(D) ← LST(D) – EST(D)

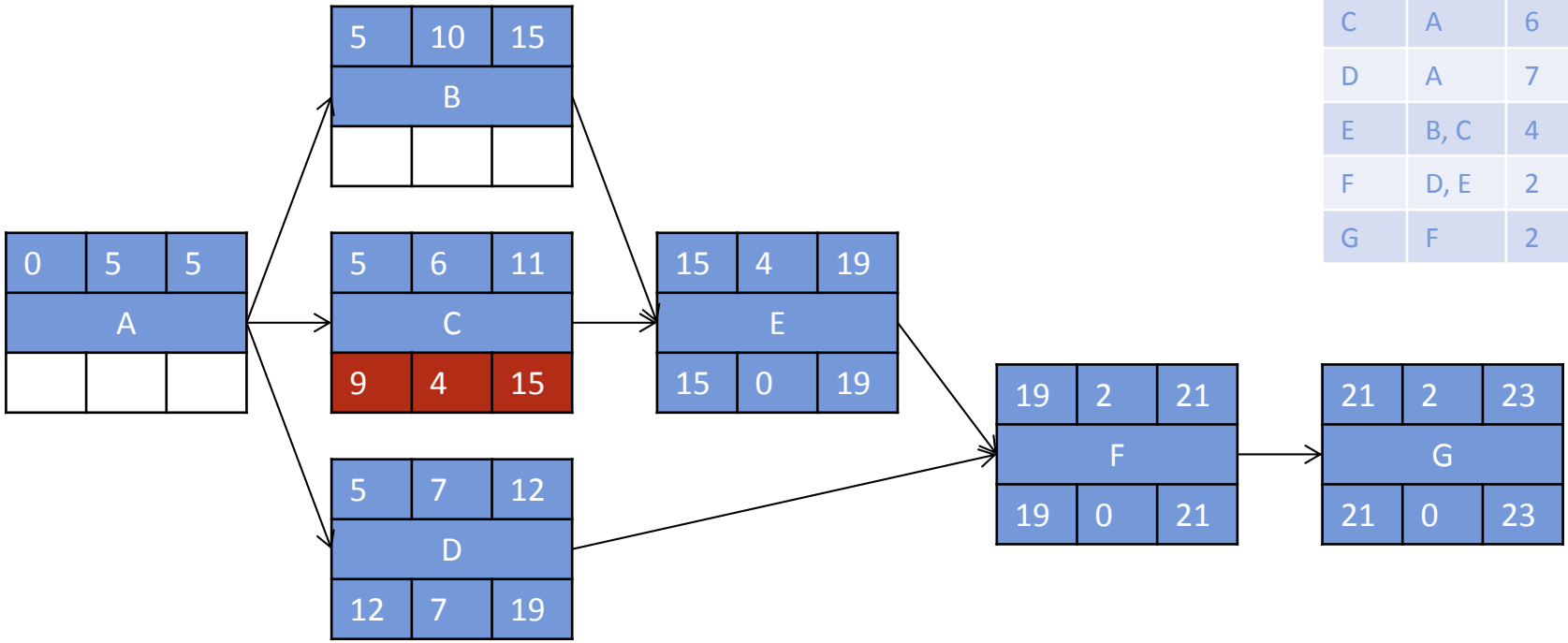
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

LFT(C) ← LST(E)
LST(C) ← LFT(C) – DUR(C)
SLACK(C) ← LST(C) – EST(C)

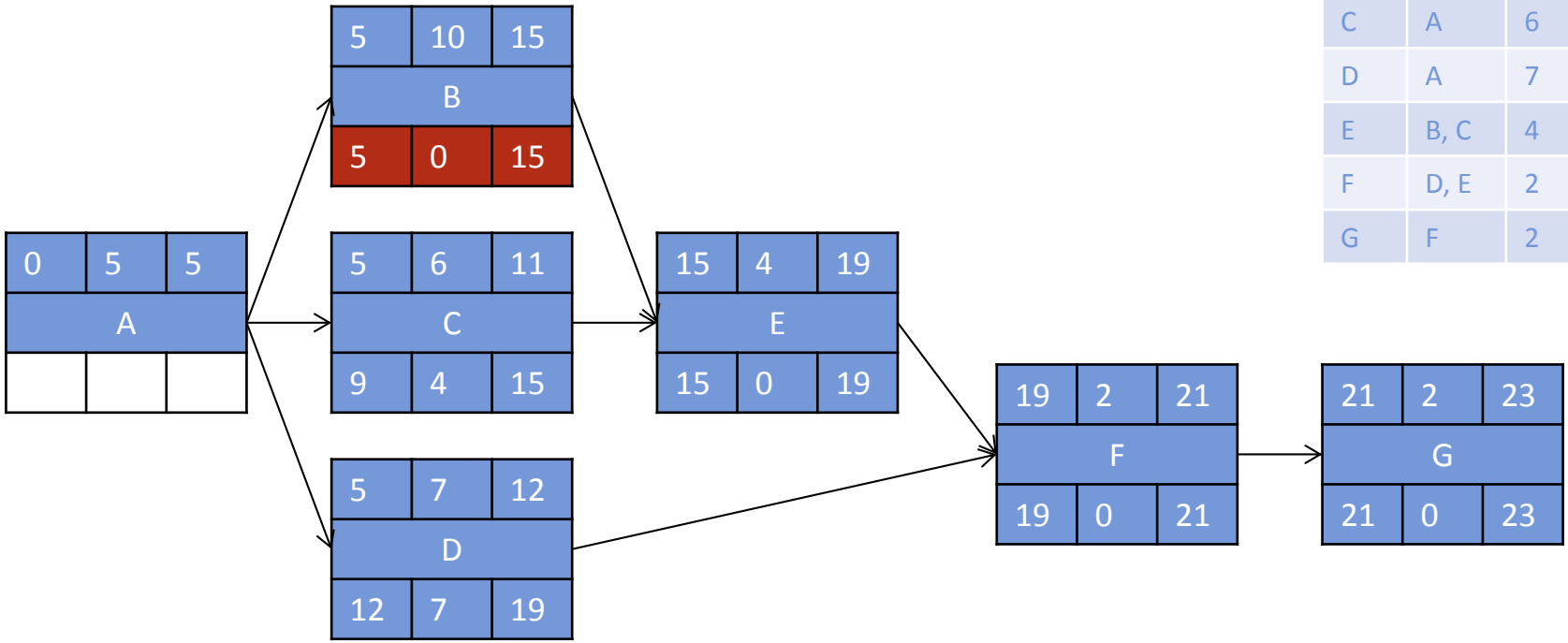
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

LFT(B) ← LST(E)
LST(B) ← LFT(B) – DUR(B)
SLACK(B) ← LST(B) – EST(B)

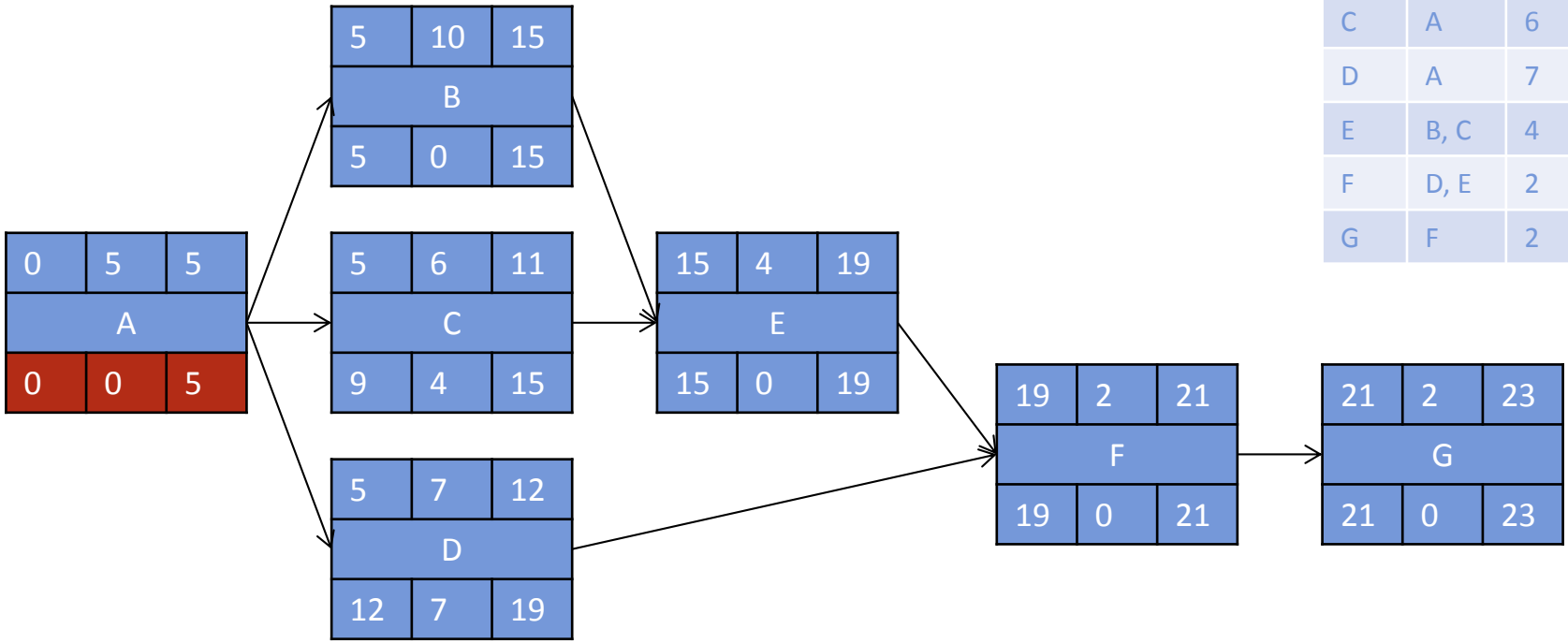
Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

LFT(A) ← MIN(LST(B), LST(C), LST(D))
LST(A) ← LFT(A) – DUR(A)
SLACK(A) ← LST(A) – EST(A)

Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2

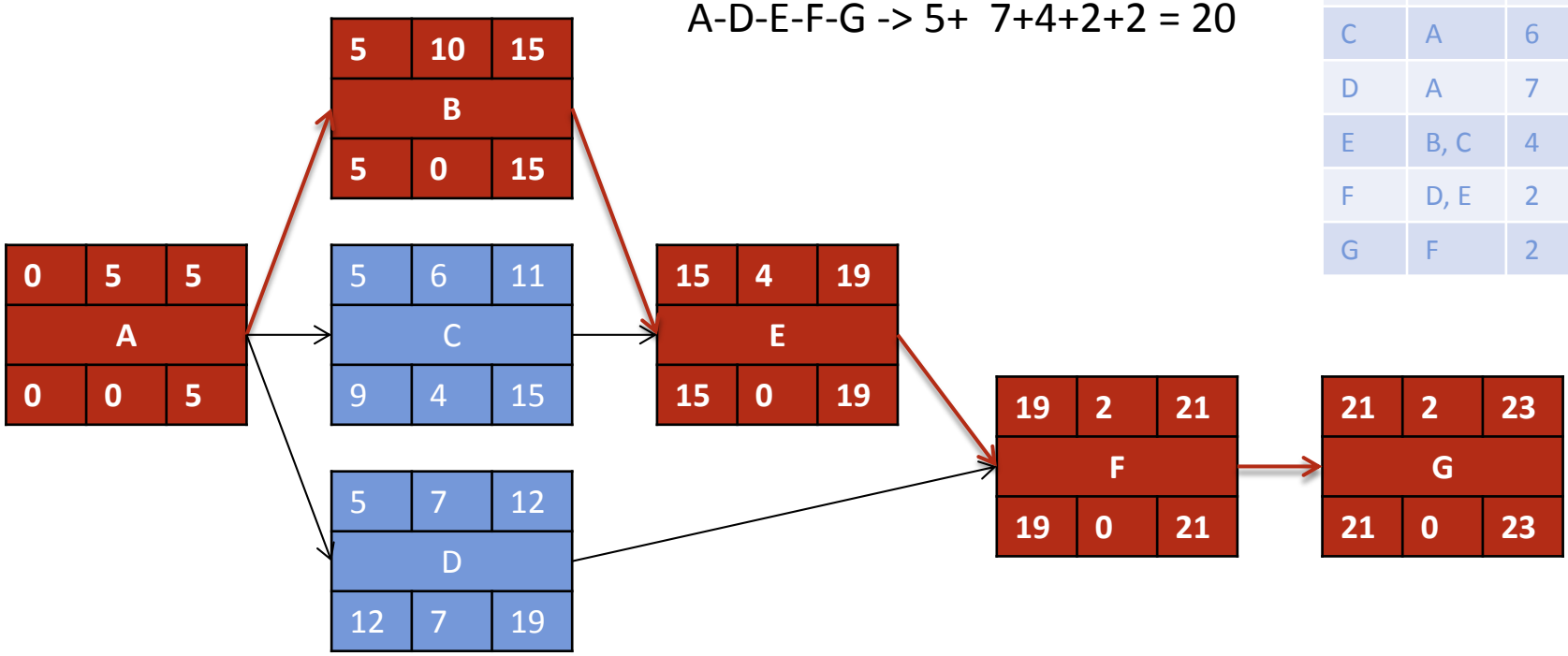


Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

In the critical path, the slack is always 0 and the sum of durations is maximized

A-B-E-F-G -> 5+10+4+2+2 = 23
A-C-E-F-G -> 5+ 6+4+2+2 = 19
A-D-E-F-G -> 5+ 7+4+2+2 = 20

Act	Pred	Dur
A	-	5
B	A	10
C	A	6
D	A	7
E	B, C	4
F	D, E	2
G	F	2



Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

Compute the critical path

Act	Pred	Dur
A	-	2
B	-	3
C	-	4
D	A	1
E	B	2
F	B	5
G	C	7
H	D, E	2
I	F, G	3
J	H, I	1

How much time do we need for this project?

Act	Pred	Dur
A	-	7
B	A	10
C	A	12
D	B	3
E	B	8
F	B	7
G	F	2
H	D,E	5
I	C	13
J	C	6
K	H	14
L	E,F	5